

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСОБІВ

ОЦІНЮВАННЯ ВПЛИВУ ШКІДЛИВИХ РЕЧОВИН

НА ЗДОРОВ'Я ЛЮДЕЙ

RESEARCH AND SOFTWARE IMPLEMENTATION OF TOOLS

TO ASSESS THE IMPACT OF HARMFUL SUBSTANCES

ON HUMAN HEALTH

Виконав: студент 2 курсу, групи КНТ-212м

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

БЄЛІКОВ О.Ю.

(ПРИЗВИЩЕ та ініціали)

Керівник ЗАЙКО Т.А.

(ПРИЗВИЩЕ та ініціали)

Рецензент КАЗУРОВА А.Є.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

БЄЛІКОВА Олексія Юрійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів оцінювання впливу шкідливих речовин на здоров'я людей. Research and Software Implementation of Tools to Assess the Impact of Harmful Substances on Human Health

керівник проєкту (роботи) к.т.н., доцент ЗАЙКО Тетяна Анатоліївна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “18” грудня 2023 року № 511

2. Строк подання студентом проєкту (роботи) 26 грудня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЗАЙКО Т.А., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “16” жовтня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки		
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	8 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	9-10 тижні	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Олексій БЄЛІКОВ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Тетяна ЗАЙКО
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
92 с., 29 рис., 3 табл., 3 дод., 19 джерел.

PYTHON, PYCHARM, МОДЕЛЬ, НЕЙРОННІ МЕРЕЖІ,
ОЦІНЮВАННЯ ЯКОСТІ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ШТУЧНИЙ
ІНТЕЛЕКТ.

Об'єкт дослідження – процес розробки програмного забезпечення для
екологічного моніторингу.

Предмет дослідження – програмні засоби для оцінювання якості
повітря.

Метою роботи є дослідження та розробка програмного забезпечення
для оцінювання впливу шкідливих речовин на здоров'я людей.

Матеріали, методи та технічні засоби: мова програмування Python,
середовище розробки PyCharm.

Результати. Розроблено програмне забезпечення для оцінювання
впливу шкідливих речовин на здоров'я людей за допомогою мови
програмування Python та середовища розробки PyCharm.

Практична цінність роботи полягає у розробці програмного
забезпечення, що та дозволяє розв'язувати завдання оцінювання впливу
шкідливих речовин на здоров'я людей.

Висновки. Виконано проєктування програмного забезпечення для
оцінювання впливу шкідливих речовин на здоров'я людей. Розроблено
програмне забезпечення для оцінювання впливу шкідливих речовин на
здоров'я людей. Здійснено тестування розробленого програмного
забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

Галузь використання – програмні засоби для оцінювання якості
повітря.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 92 pages, 29 figures, 3 tables, 3 appendixes, 19 sources.

PYTHON, PYCHARM, MODEL, NEURAL NETWORKS, QUALITY ASSESSMENT, SOFTWARE, ARTIFICIAL INTELLIGENCE.

The object of research is the process of developing software for environmental monitoring.

The subject of the research is software for air quality assessment.

The purpose of this work is to develop the software for assessing the impact of harmful substances on human health.

Materials, methods and technical tools: Python programming language, PyCharm development environment.

Results. The software tools for assessing the impact of harmful substances on human health using Python programming language and PyCharm development environment have been developed.

The practical value of the work lies in the development of software that allows solving the task of assessing the impact of harmful substances on human health.

Conclusions. The software for assessing the impact of harmful substances on human health has been designed. The software for assessing the impact of harmful substances on human health has been developed. The developed software for assessing the impact of harmful substances on human health has been tested.

Scope of use – software tools for air quality assessment.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Вплив шкідливих речовин на здоров'я людей.....	11
1.2 Аналіз програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.....	19
1.3 Висновки за розділом 1	33
2 Матеріали і методи	35
2.1 Вибір мови програмування	35
2.2 Вибір середовища розробки для розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей	38
2.3 Методи та моделі машинного навчання для оцінювання впливу шкідливих речовин	40
2.4 Розробка моделі програмного забезпечення для оцінювання впливу шкідливих речовин	46
2.5 Висновки за розділом 2	48
3 Опис програми	49
3.1 Структура програмного забезпечення для оцінювання впливу шкідливих речовин.....	49
3.2 База даних програмного забезпечення для оцінювання впливу шкідливих речовин	52
3.3 Функціонування програмного забезпечення для оцінювання впливу шкідливих речовин	54
3.4 Проектування інтерфейсу програмного забезпечення для оцінювання впливу шкідливих речовин	56
3.5 Висновки за розділом 3	58
4 Експлуатація, тестування та експериментальне дослідження програми.....	59
4.1 Призначення й умови застосування програми	59

4.2	Характеристики програми для оцінювання впливу шкідливих речовин на здоров'я людей.....	60
4.3	Інструкція по експлуатації програми.....	61
4.3.1	Звернення до програми.....	61
4.3.2	Вхідні й вихідні дані	61
4.3.3	Повідомлення.....	62
4.4	Виконання програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей	62
4.5	Тестування програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей	65
4.6	Висновки за розділом 4	65
	Висновки	66
	Перелік джерел посилання	69
	Додаток А Технічне завдання	71
	Додаток Б Фрагмент тексту програми	75
	Додаток В Слайди презентації	85

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – Application Programming Interface;

IDE – Integrated Development Environment;

MLP – Multilayer Perceptron;

ГА – Генетичний алгоритм;

НМ – нейронні мережі;

ПЗ – програмне забезпечення.

ВСТУП

Використання програмних засобів для оцінювання впливу шкідливих речовин в повітрі на здоров'я людей залишається вкрай актуальним. Завдяки таким засобам можна аналізувати рівень забруднення повітря, визначати концентрацію токсичних речовин та їх вплив на здоров'я. Це дозволяє урядовим установам, організаціям охорони здоров'я та громадськості реагувати на шкідливі викиди, впроваджувати заходи для зменшення забруднення та забезпечувати захист громадського здоров'я. Таке програмне забезпечення створює можливість отримання точних та достовірних даних про якість повітря та сприяє управлінню екологічною ситуацією для покращення якості життя населення [1]-[6].

Програмні засоби для оцінювання впливу шкідливих речовин в повітрі є важливим інструментом для визначення рівня забруднення атмосфери та його впливу на здоров'я людей. Ці програмні рішення зазвичай використовуються для моніторингу та аналізу різноманітних параметрів повітря, таких як концентрація токсичних речовин, рівень забруднення, та інші параметри, які можуть мати негативний вплив на здоров'я. Вони допомагають у визначенні якості повітря, можливих джерел забруднення, відслідковують тенденції змін у рівнях різних речовин, а також надають аналіз даних, який сприяє прийняттю рішень у зв'язку з мінімізацією шкідливого впливу на довкілля та здоров'я людей. Такі засоби мають значний вплив на формування стратегій екологічного управління та забезпечення безпеки громадського здоров'я [1]-[6].

Проте існуючі програмні засоби не завжди можуть забезпечувати точні та достовірні результати оцінювання впливу шкідливих речовин на здоров'я людей, особливо у випадках, коли мережа датчиків недостатня або у вимірюванні домінує певна погрішність. Крім того, програмне та апаратне забезпечення не завжди може охопити усі області або забезпечити однакову точність результатів в усіх регіонах, що ускладнює загальне оцінювання стану

повітря. Деякі програмні засоби можуть бути високо вартісними, особливо якщо потрібно розширення функціональності або доступ до більш деталізованих даних [1]-[6].

Тому актуальною є розробка програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для оцінювання впливу шкідливих речовин на здоров'я людей;
- розробити розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей;
- здійснити проєктування програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей;
- створити програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей;
- виконати тестування розробленого програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Вплив шкідливих речовин на здоров'я людей

Шкідливі речовини, які можна знайти у повітрі, охоплюють широкий спектр забруднюючих речовин. Деякі з них включають в себе оксиди азоту, сірководніди, вуглеводні, бензопірен, а також важкі метали, такі як свинець, ртуть та хром. Ці речовини можуть бути викинуті в результаті промислової діяльності, автотранспорту, теплових станцій та інших джерел, що призводять до забруднення повітря. Вони можуть мати негативний вплив на здоров'я людей, спричиняючи різноманітні захворювання дихальних шляхів, серцево-судинну систему та інші проблеми зі здоров'ям [1]-[3].

У зв'язку із зростанням кількості промислових підприємств, які викидають різноманітні забруднюючі речовини в атмосферу, проблема забруднення повітря стала особливо актуальною. Токсини, які впливають на оточуюче середовище, набувають різноманітних форм. Можна виділити природні та антропогенні джерела забруднення повітря [1]-[3].

Природні джерела забруднення атмосфери включають різні погодні явища, такі як пилові бурі, виверження вулканів, космічний пил і т. д. Загалом, близько трьох чвертей природного забруднення повітря становлять речовини неорганічного походження – частинки ґрунту, каміння. Продукти, що відносяться до них, витримують атмосферні умови, такі як пил, неорганічні солі та інші [1]-[6].

Атмосферне повітря містить різноманітні органічні речовини, що є результатом життєдіяльності різних організмів. Це можуть бути вуглеводи, органічні кислоти, прості ефіри, альдегіди та інші маломолекулярні органічні сполуки. Ці речовини використовуються багатьма живими організмами для їхніх потреб. Деякі з цих органічних сполук, шкідливі для найпростіших мікроорганізмів, бактерій та грибів, називаються фітонцидами. У атмосферу щороку потрапляє велика кількість морської солі та ґрунтового пилу. Наприклад, під час лісових пожеж утворюються частки горючих речовин та

твердих часток, що утворюють аерозолі. Ці аерозолі становлять велику частку складу природних забруднень в атмосфері. Хоча природні джерела забруднення в цілому не є серйозними проблемами для людини, вони можуть викликати значні зміни в умовах оточуючого середовища на великій території. У певних ситуаціях такі процеси можуть призвести до утворення непрозорого шару, що оточує планету, що може впливати на тепловий баланс. Тим не менш, природні джерела забруднення повітря в основному знаходяться під впливом біологічних законів, що дозволяє їм не викликати серйозних проблем для людини [1]-[6].

Штучне або антропогенне забруднення атмосфери — це зміна складу та властивостей середовища через діяльність людини. Джерела цього виду забруднення можна поділити на дві основні категорії: стаціонарні та рухомі. Залежно від джерела, забруднення повітря можна класифікувати як фізичне або хімічне. Фізичне забруднення включає пил, механічні частинки, радіаційне випромінювання, електромагнітне поля, шум та нагрівання [1]-[6].

Усі джерела техногенного забруднення поділяються на стаціонарні, які постійно викидають забруднення в атмосферу (це можуть бути підприємства чи заводи), та рухомі, що розносять забруднення у просторі (наприклад, транспортні засоби). Стаціонарні джерела можна поділити на організовані (які мають чітко визначені джерела викидів) та розпорошені (де викиди розподілені на різні джерела) [1]-[6].

Організовані джерела — це технічні системи, через які здійснюється викид забруднюючих речовин, такі як труби, газопроводи та інші технічні споруди. Нескоординовані викиди забруднюючих речовин у повітря зазвичай виникають раптово та без контролю, наприклад, через витік обладнання, порушення технічних норм, відсутність очисних систем або умови праці, що не відповідають вимогам. Це можуть бути складські споруди, місця вантажно-розвантажувальних операцій. Також до неструктурованих джерел можуть відноситися парковки, зберігання паливно-мастильних матеріалів та інші не пов'язані з енергетикою джерела [1]-[6].

Склад викидів в атмосферу від стаціонарних джерел забруднення головним чином залежить від типу речовини та технології, які використовуються цими джерелами. Наприклад, теплові електростанції викидають оксиди сірки, оксиди азоту, сажу, пил та інші речовини, що утворюють сірчану кислоту при контакті з водою, тверді частки та металеві солі. Вуглекислий газ, оксиди сірки, сірчистий ангідрид, пил, оксиди азоту, сірководень, аміак, сірковуглець, хром, марганець, бензол, фенол та інші органічні сполуки також можуть присутні у викидах. Для підприємств, що займаються металургією, у викидах можуть бути сполуки кольорових і важких металів, фтор, пари ртуті, оксиди сірки, оксиди вуглецю, смолисті матеріали, поліметалевий пил, вуглеводні та інші органічні сполуки низької молекулярної маси. Однією з найнебезпечніших речовин, що містяться у викидах підприємств металургії, є бензпірен, який має канцерогенну властивість та може негативно впливати на здоров'я людини [1]-[6].

Виробництво в металургії та машинобудуванні випускає речовини у повітря, схожі на викиди. Ці викиди складаються з часток забруднюючих речовин, таких як фарбові сполуки та важкі метали (включаючи пари ртуті). Компанії, що займаються нафтохімією та видобутком нафти, також викидають різні речовини, такі як сірководень, сірку, діоксид сірки, різновиди вуглецевих оксидів, аміак, сполуки низькомолекулярних та гетероциклічних вуглеводнів (особливо бензпірен) [1]-[6].

У хімічній промисловості також існують небезпечні речовини, такі як сіркові оксиди (особливо сірка та діоксид сірки), хлор, оксиди азоту, аміак, сірководень, фосфорні сполуки, органічні з'єднання, важкі метали, які утворюють хлориди, пил та пісок [1]-[6].

Виготовлення оздоблювальних та будівельних матеріалів супроводжується викидами у повітря пилу, діоксиду кремнію, гіпсового та азбестового пилу, фтору та сполук важких металів [1]-[6].

Забруднення повітря внаслідок руху транспорту: рух на дорогах є ключовим джерелом забруднення повітря. Загалом український транспорт, що

пересувається, відповідає за приблизно тридцять вісім відсотків забруднення повітря [1]-[6].

Якість викидів залежить від типу та якості палива, що використовується у двигуні, особливостей його конструкції та умов експлуатації. При повному згоранні палива (за оптимальних умов роботи двигуна) зазвичай утворюються вуглекислий газ, вода (або водяна пара) та діоксид сірки. Якщо кисню недостатньо, вуглеводні, які становлять більшу частину палива, не встигають повністю згорати. У такому випадку утворюється оксид вуглецю, що є дуже отруйним і може спричинити серйозне отруєння навіть при низьких концентраціях у повітрі. Зазвичай вихлопні гази містять близько двохсот різних компонентів, які можуть розкладатися у природних умовах від кількох хвилин до кількох років [1]-[6].

Усі речовини, які забруднюють повітря, у різний спосіб впливають на здоров'я людини. Ці речовини потрапляють у тіло переважно через дихальні шляхи, шкіру та слизові оболонки. Забруднене повітря безпосередньо впливає на роботу дихальної системи. Частки дуже невеликим розміром можуть потрапити у заражені легені через дихання, і до половини цих часток може залишитися там. Токсичні наслідки можуть виникати з таких причин [1]-[6]: речовини є токсичними через їхні хімічні або фізичні властивості; речовини можуть заважати нормальним процесам очищення дихальних шляхів; речовини можуть поглинати інші токсини та виділяти їх у дихальних шляхах до легенів [1]-[6].

Іноді забруднюючі речовини, які потрапляють в організм людини, можуть підсилювати токсичні властивості інших речовин. Таким чином, загальний токсичний вплив декількох таких агентів може перевищити їхні індивідуальні ефекти. Також важливо враховувати тривалість і ступінь токсичного впливу отруйних речовин [1]-[6].

Статистичні дослідження використовуються для виявлення зв'язку між рівнем забруднення атмосферного повітря та виникненням хвороб, таких як респіраторні захворювання, проблеми з очима та інші проблеми здоров'я.

Підвищення рівня забруднення повітря протягом кількох днів призводить до збільшення смертності серед людей похилого віку від захворювань дихальної системи та серцево-судинних захворювань [1]-[6].

У випадку професійних захворювань, наприклад, серед працівників, які працювали у зоні впливу азбесту, спостерігали рак легень і діафрагми (м'яз, що розділяє грудну порожнину від черевної). Довготривалий контакт з берилієм збільшує ризик онкологічних захворювань дихальної системи, шкіри та очей [1]-[6].

Пари ртуті є надзвичайно отруйними та можуть спричиняти порушення центральної нервової системи та нирок, що відомо як ртутний нирковий синдром. Також важливо пам'ятати, що ртуть може накопичуватися в організмі людини, і отруєння нею може мати кумулятивний ефект [1]-[6].

У містах значно зростає викид забруднюючих речовин, що стає серйозною проблемою. Це призводить до збільшення числа хвороб у більшості розвинених країн і в Україні, таких як хронічний бронхіт, емфізем, алергії та онкологічні захворювання [1]-[6].

Отруєння виникає, коли невеликі дози шкідливих речовин систематично або періодично потрапляють в організм. Симптоми серйозного отруєння включають порушення поведінки, нервово-психічні розлади та зміни у стані: швидка втомлюваність, виснаженість, безсоння, депресія, неуважність, апатія, проблеми з пам'яттю та раптові зміни настрою [1]-[6].

Отруєння повітряно-крапельне може бути гострим або хронічним. Гостре отруєння зазвичай виникає при високій концентрації речовини в повітрі та короткому тривалому впливі. Довготривалий контакт з токсичними речовинами може призвести до хронічного отруєння [1]-[6].

Те, як організм реагує на отруйні речовини, залежить від особливостей кожної людини. Тому одна й та ж отруйна речовина може викликати різні захворювання у різних людей. Подібні закономірності спостерігаються й у радіоактивному забрудненні повітря [1]-[6].

Хімічні сполуки, які мають високу біологічну активність, можуть призвести до серйозних уражень організму людини, спричиняючи системні порушення, вплив на нервову систему та негативний вплив на формування плоду та розвиток дитини [1]-[6].

Дослідження показують пряму залежність між зростанням алергічних реакцій, бронхіальної астми та онкологічних захворювань і погіршенням екологічної ситуації на певній території. Компоненти, такі як хром, нікель, берилій та азбест, відомі як доказані канцерогени [1]-[6].

Рак у дітей, який колись був рідкістю у першій половині двадцятого століття, зараз стає все більш поширеним. Також зафіксовані нові, раніше невідомі захворювання [1]-[6]. Усюди в Україні спостерігається схожий тренд у збільшенні викидів забруднюючих речовин. За останні дві тисячі років кількість таких речовин, що потрапляють у повітря, безперервно зростала, а на деяких територіях це зростання було надзвичайно великим. Взагалі викиди продовжують збільшуватися на більшості території України. У деяких областях рівень викидів залишається стабільним, коливаючись в межах спостережуваних показників за дві тисячі років [1]-[6].

В Україні основним чинником, що впливає на ступінь забруднення повітря та викидів вуглекислого газу стаціонарними джерелами, є економічна екологічна ситуація в регіоні та в країні загалом [1]-[6]. З цієї інформації можна припустити, що існує зв'язок між складом викидів в атмосферу та системами дихання та кровообігу, а також між обсягом викидів та рівнем захворювань, зокрема онкологічних захворювань. Особливо це стосується легеневих хвороб, де зв'язок з викидами виявляється більш очевидно, а вплив на систему кровообігу стає більш конкретним [4]-[7].

Останнім часом зростання туберкульозу стало предметом уваги громадськості. Широке поширення цієї хвороби розглядається як індикатор соціального стану. Туберкульоз часто впливає на людей із низьким рівнем доходу та соціальним статусом, а загальне зростання захворювання може бути пояснене зменшенням державних доходів та економічними труднощами,

пов'язаними з цими факторами [1]-[6]. Проте важливо пам'ятати, що в контексті туберкульозу легенів не можна ігнорувати рівень забруднення повітря, спостерігаючи за динамікою серед пацієнтів [1]-[6].

Серед факторів, які можуть сприяти цьому становищу, фахівці вказують на слабкість імунітету, хронічні захворювання дихальних шляхів і вплив постійних подразників на слизові оболонки [1]-[6].

Доведено, що існує зв'язок між туберкульозом і підвищеним ризиком виникнення раку легень. Туберкульоз залишає в легенях окремі ділянки з вогнищевими змінами (своєрідні рубці), що може значно підвищити ймовірність раку легень [2]-[7].

У зв'язку з цим важливим є дослідження впливу забруднення повітря та викидів вуглекислого газу на розвиток хвороб системи кровообігу, таких як туберкульоз та рак [1]-[6].

Використання програмних засобів для оцінювання впливу шкідливих речовин в повітрі на здоров'я людей залишається вкрай актуальним. Завдяки таким засобам можна аналізувати рівень забруднення повітря, визначати концентрацію токсичних речовин та їх вплив на здоров'я. Це дозволяє урядовим установам, організаціям охорони здоров'я та громадськості реагувати на шкідливі викиди, впроваджувати заходи для зменшення забруднення та забезпечувати захист громадського здоров'я. Таке програмне забезпечення створює можливість отримання точних та достовірних даних про якість повітря та сприяє управлінню екологічною ситуацією для покращення якості життя населення. Програмні засоби для оцінювання впливу шкідливих речовин в повітрі є важливим інструментом для визначення рівня забруднення атмосфери та його впливу на здоров'я людей. Ці програмні рішення зазвичай використовуються для моніторингу та аналізу різноманітних параметрів повітря, таких як концентрація токсичних речовин, рівень забруднення, та інші параметри, які можуть мати негативний вплив на здоров'я. Вони допомагають у визначенні якості повітря, можливих джерел забруднення, відслідковують тенденції змін у рівнях різних речовин, а також надають аналіз

даних, який сприяє прийняттю рішень у зв'язку з мінімізацією шкідливого впливу на довкілля та здоров'я людей. Такі засоби мають значний вплив на формування стратегій екологічного управління та забезпечення безпеки громадського здоров'я [4]-[6].

Отже важливим завданням є дослідження рівня захворюваності на хвороби серця, туберкульозу і онкологічних захворювань у залежності від рівнів забруднення повітря від промислових джерел у різних регіонах. Для розв'язання цього завдання важливим є застосування методів та програмних засобів, що дозволять досліджувати зв'язки між захворюваністю та кількістю викидів, що може бути використане для прогнозування показників захворюваності та включено до планування відповідних дій по зниженню рівнів захворюваності населення [1]-[7].

Дослідження у цих сферах стали актуальними та привернули увагу світових наукових спільнот в галузі медицини та інформатики. Згідно зі статутом Всесвітньої організації охорони здоров'я, забезпечення найвищого рівня здоров'я стало одним з основних прав людини, а охорона навколишнього середовища, особливо повітря та води, є одним із ключових завдань [1]-[6].

Моделювання цієї динаміки потребує використання складних інструментів, таких як штучні нейронні мережі та генетичні алгоритми. Важливо проаналізувати зміни в рівнях забруднення атмосфери від стаціонарних джерел і їх вплив на туберкульоз, захворювання серцево-судинної системи та онкологічні захворювання в різних регіонах. Для дослідження такого впливу важливо використовувати не тільки традиційні математичні моделі, але й більш складні техніки, такі як штучні нейронні мережі та генетичні алгоритми, після чого доцільно визначено оптимальну модель, що залежить від показників захворюваності, та на основі синтезованої моделі розробити відповідне програмне забезпечення для прогнозування рівня захворюваності на окремі хвороби [1]-[7].

Відзначено, що у наш час шкідливі речовини, які можна знайти у повітрі, охоплюють широкий спектр забруднюючих речовин. Проаналізовано

фактори, які пов'язані з рівнем забруднення повітря та можуть впливати на рівні захворюваності населення. Підкреслено, що використання програмних засобів для оцінювання впливу шкідливих речовин в повітрі на здоров'я людей залишається вкрай актуальним. Завдяки таким засобам можна аналізувати рівень забруднення повітря, визначати концентрацію токсичних речовин та їх вплив на здоров'я. Це дозволяє урядовим установам, організаціям охорони здоров'я та громадськості реагувати на шкідливі викиди, впроваджувати заходи для зменшення забруднення та забезпечувати захист громадського здоров'я. Таке програмне забезпечення створює можливість отримання точних та достовірних даних про якість повітря та сприяє управлінню екологічною ситуацією для покращення якості життя населення. Програмні засоби для оцінювання впливу шкідливих речовин в повітрі є важливим інструментом для визначення рівня забруднення атмосфери та його впливу на здоров'я людей. Ці програмні рішення зазвичай використовуються для моніторингу та аналізу різноманітних параметрів повітря, таких як концентрація токсичних речовин, рівень забруднення, та інші параметри, які можуть мати негативний вплив на здоров'я. Вони допомагають у визначенні якості повітря, можливих джерел забруднення, відслідковують тенденції змін у рівнях різних речовин, а також надають аналіз даних, який сприяє прийняттю рішень у зв'язку з мінімізацією шкідливого впливу на довкілля та здоров'я людей [1]-[7].

1.2 Аналіз програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей

Проаналізуємо програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей [8]-[12].

Програмне забезпечення AirQ+ є програмним інструментом, розробленим Європейським регіональним офісом Всесвітньої організації охорони здоров'я, який визначає вплив та навантаження поведінки, пов'язаної

з відходами, на навколишнє середовище. Програмне забезпечення AirQ+ (рис. 1.1) включає у себе інструменти для вимірювання короткотермінового та довготермінового впливу забруднювачів на навколишнє середовище. Основні методи базуються на даних когортних епідеміологічних досліджень, щоб показати зв'язок між середніми рівнями концентрації забруднювачів у повітрі та ризиком смертності у популяції під час тривалих періодів часу. Оцінка наслідків забруднення повітря проводиться програмним забезпеченням AirQ+ під час аналізу політик та втручань або результатів гіпотетичних сценаріїв [8]-[10].

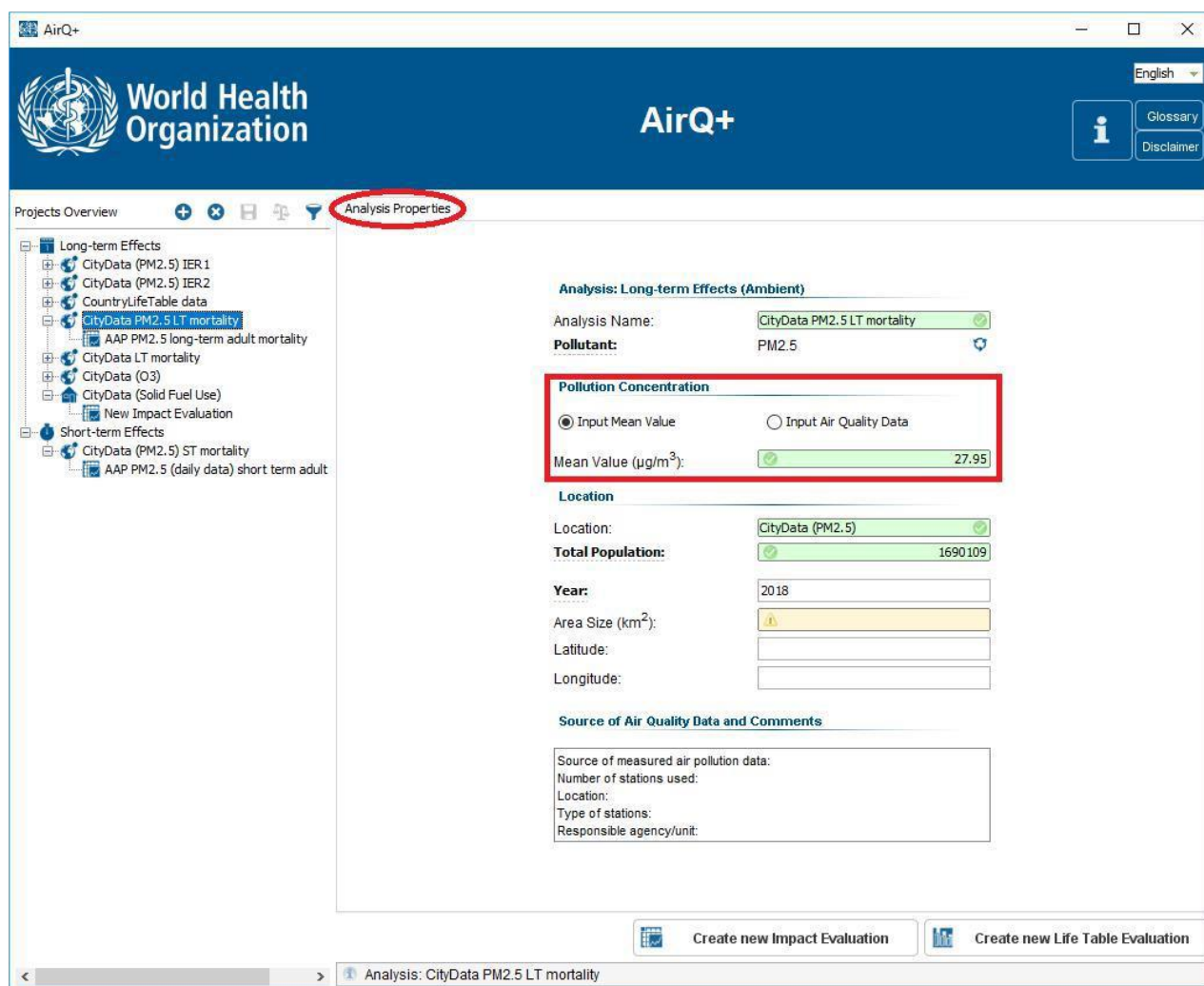


Рисунок 1.1 – Програмне забезпечення AirQ+ [9]

Програмне забезпечення AirQ+ рекомендується використовувати спільно з епідеміологом або фахівцем з забруднення повітря. Щоб спростити

процес аналізу, AirQ+ має посібник користувача, який потребує додаткових знань. Програмне забезпечення AirQ+ може розраховувати ризик на основі окремих ризиків, одержаних з професійних досліджень або даних про токсичність [8]-[10].

Користувачі, які мають інформацію про концентрації певних речовин у повітрі, але не мають інформації про традиційні забруднювачі, можуть використовувати AirQ+ для оцінки ризику раку на основі окремих значень ризику. AirQ+ використовує стандартні значення ризику для миш'яку, бензолу, хрому, нікелю та вінілхлориду [8]-[10].

Програмне забезпечення AirQ+ розроблено для оцінки впливу якості повітря та впливу місцевості на здоров'я. Це здійснюється через аналіз даних, графічні засоби, таблиці та кількісні дані про основні забруднювачі, такі як тверді частки, діоксид азоту і тропосферний озон. Airq+ також може оцінити сажу та визначити вплив повітря всередині приміщень на здоров'я. Цей інструмент можна використовувати для оцінки якісного стану повітря та довготривалого забруднення приміщень, спричиненого використанням твердого палива [8]-[10].

AirQ+ має інструменти для оцінки впливу забруднювачів на середовище короткостроково та довгостроково. Він використовує дані когортних епідеміологічних досліджень, щоб показати зв'язок між середніми рівнями забруднюючих речовин у повітрі та ризиком смертності серед популяцій, що піддавалися впливу тривалий час. Оцінка наслідків забруднення повітря проводиться при оцінці політики, втручань або результатів гіпотетичних сценаріїв [8]-[10].

Для користування AirQ+ важливо використовувати його спільно з епідеміологом або фахівцем у сфері забруднення повітря. Щоб спростити аналіз користувача, AirQ+ має посібник, який вимагає додаткового досвіду в області використання цього інструменту [8]-[10].

Серед особливостей програмного забезпечення AirQ+ можна виділити такі [8]-[10]:

– засоби аналізу якості повітря: програмне забезпечення AirQ+ забезпечує оцінку рівнів різних забруднювачів повітря, таких як тверді частки, діоксид азоту, тропосферний озон, сажа та інші, що дозволяє користувачеві оцінити вплив цих забруднень на здоров'я [8]-[10];

– засоби оцінювання ризиків для здоров'я: програмне забезпечення AirQ+ розраховує короткостроковий і довгостроковий вплив забруднювачів на здоров'я людей, враховуючи дані когортних епідеміологічних досліджень. Це дозволяє оцінити ризик смертності та інші показники здоров'я у залежності від рівнів забруднень повітря [8]-[10];

– засоби підтримки експертів: AirQ+ рекомендується використовувати спільно з фахівцями з епідеміології та забруднення повітря, оскільки для правильної інтерпретації результатів важливо мати додатковий експертний контекст [8]-[10];

– зручний посібник користувача: даний інструмент постачається з посібником, який допомагає користувачам розуміти та застосовувати AirQ+ для оцінки ризиків для здоров'я, особливо щодо впливу забруднень повітря на ризик виникнення захворювань, таких як рак [8]-[10];

– розширені можливості аналізу: окрім стандартних методів, програма використовує різні інструменти, включаючи алгоритми рою частинок та штучні нейронні мережі для більш точної оцінки впливу забруднень повітря на здоров'я [8]-[10];

– засоби оцінювання різноманітності забруднень: програма враховує широкий спектр забруднювачів, не обмежуючись лише загальноновживаними. Вона дозволяє враховувати токсичні речовини, які можуть бути менш поширеними, але мають суттєвий вплив на якість повітря [8]-[10];

– географічна пристосованість: AirQ+ розрахований на роботу в різних географічних зонах та місцевих умовах, що дає можливість використовувати його для оцінки впливу забруднень повітря у різних країнах та регіонах з різними стандартами якості повітря [8]-[10];

- динамічність досліджень: програмне забезпечення AirQ+ орієнтовано на актуальність даних і дозволяє проводити динамічні аналізи змін впливу забруднювачів на здоров'я населення в залежності від розвитку ситуації з якістю повітря [8]-[10];

- універсальність застосування: AirQ+ можна використовувати для різних цілей: від оцінки загального впливу забруднень повітря на населення до аналізу конкретних забруднювачів та їх впливу на здоров'я [8]-[10];

- допоміжні інструменти: крім аналізу забруднень, програма може надавати рекомендації щодо покращення якості повітря та зменшення його впливу на здоров'я, що робить її корисним інструментом для розробки політики екологічного управління [8]-[10].

Серед недоліків програмного забезпечення AirQ+ можна відзначити такі [8]-[10]:

- обмежена точність: програмне забезпечення AirQ+ може мати обмежену точність при прогнозуванні впливу забруднень на здоров'я, оскільки залежить від доступної інформації та методів розрахунків [8]-[10];

- потреба в даних: робота програми програмного забезпечення AirQ+ вимагає наявності відповідних даних про забруднення повітря, які можуть бути обмеженими або відсутніми для деяких регіонів чи періодів [8]-[10];

- специфічні умови застосування: методи, які використовує програмне забезпечення AirQ+ для оцінки впливу, можуть бути специфічними для певних ситуацій, що ускладнює її універсальне використання [8]-[10];

- складність інтерпретації результатів: отримані результати можуть бути складними для інтерпретації для осіб без спеціалізованого фахового знання з епідеміології або екології [8]-[10];

- необхідність експертної участі: для використання програмного забезпечення AirQ+ у реальних ситуаціях може бути необхідна експертна оцінка або участь фахівця з екології чи епідеміології для адекватної інтерпретації та використання отриманих даних [8]-[10].

Програмне забезпечення EPI Air Technology є інноваційною системою очищення повітря. В основному, повітряні системи ЕРІ використовуються у сільськогосподарських та тваринницьких спорудах, таких як птахівники або інші закриті приміщення. Встановлення EPI Air (рис. 1.2) призводить до покращення продуктивності тварин та полегшення дихання, що позитивно впливає на ваше господарство [11].

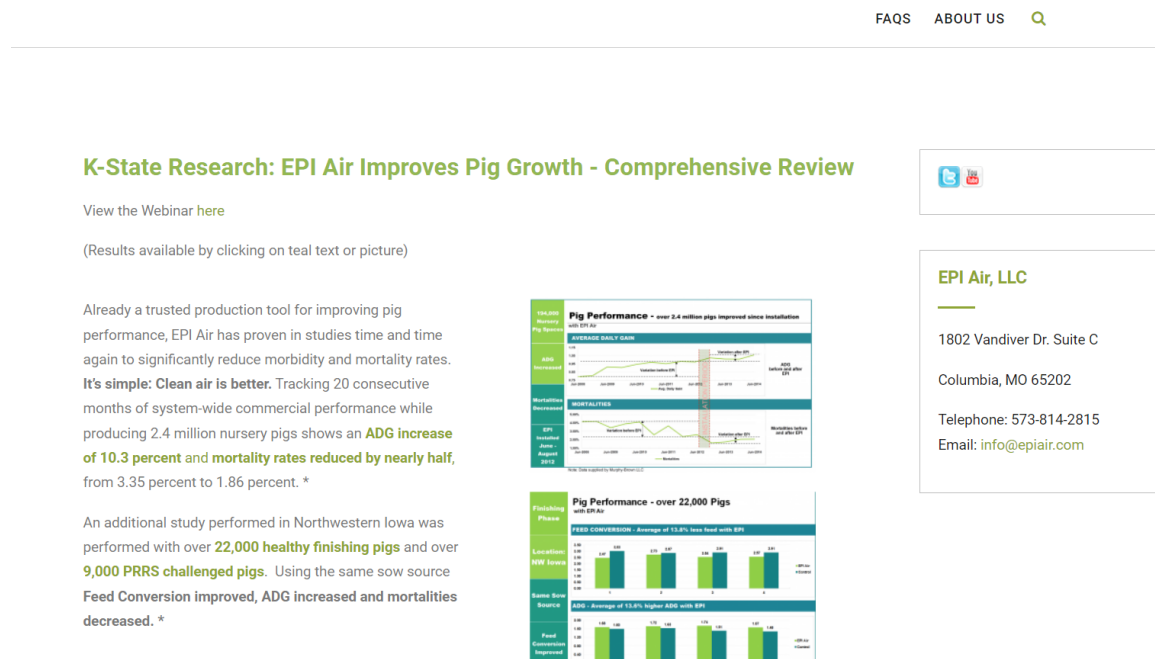


Рисунок 1.2 – Програмне забезпечення EPI Air [11]

EPI Air складається з основних елементів: джерело живлення, коронна трубка, моторизована лебідка та перемикач НТ [11]. Споживання та розумний контроль: система низьковольтного високовольтного живлення та інтелектуальний контролер пов'язані з коронним шлангом, що регулюється та керується системою для створення чистого повітря за допомогою EPI Air Technology [11]. Коронна трубка виготовлена зі сталі та встановлена під стелею. Вона регулює розподіл іонів в повітрі, прибираючи пил, бактерії та інші забруднення. Ця патентована технологія керує потоком іонів, що сприяє очищенню повітря, покращуючи дихальні шляхи тварин [11]. Вимикач високої напруги слугує для включення та вимкнення програмної системи EPI Air в процесі використання статичної електрики [11].

EPI Air є патентованою системою аналізу та очищення повітря, спеціально розробленою для зменшення пилу, бактерій і газів у сільськогосподарських та обмежених просторах. Дослідження показують позитивні зміни у виробництві та життєздатності. Проводяться дослідження для вивчення впливу на дихальні та імунні системи. Існує припущення, що ці системи зменшують стрес і сприяють кращому росту. EPI Air надає інформацію про ці дані [11].

Програмна система EPI Air System аналізує якість повітря та за необхідності розпилює в повітрі закритого середовища величезну кількість заряджених іонів, що допомагає у видаленні пилу, хвороботворних мікроорганізмів та різних газів [11]. Негативні іони та частки, які рухаються у повітрі, взаємодіють між собою. Ця взаємодія призводить до поляризації частинок рідини, яка оточує іони. Вона дає частинкам певні позитивні та негативні властивості, внаслідок чого вони поведуться подібно до магнітів, притягуючись один до одного та поверхонь, з якими вони контактують [11].

Дослідження вказують на те, що програмна система EPI Air System сприяє зниженню кількості патогенів у повітрі. EPI Air System працює з патогенами у навколишньому середовищі, навіть якщо вони присутні без інфекції, або у високих концентраціях негативних іонів, які впливають на мікробні клітини [11]. Коли пилові частинки осідають на поверхнях, повітря, яке потрапляє до тварин, стає чистішим, ніж те, що коливається та летить. Технологія EPI Air покращує якість повітря, зменшуючи пил в його джерелах. Наукові дослідження свідчать про те, що чисте повітря покращує якість життя населення [11].

Технологія EPI Air створює свіже повітря всередині приміщення, що сприяє кращим умовам, покращенню умов праці та зниженню статичної електрики для оточуючих [11].

Серед особливостей програмного забезпечення EPI Air можна виділити такі [11]:

– засоби зменшення патогенів у повітрі: програмне забезпечення EPI Air дозволяє керувати процесом підвищення якості повітря, що зменшує кількість хвороботворних мікроорганізмів у повітрі. Це реалізується шляхом автоматичного розпилення в повітря великої кількості заряджених іонів, що видаляють пил, бактерії та повітряні гази [11];

– засоби аналізу якості повітря: програмне забезпечення EPI Air забезпечує оцінку рівнів різних забруднювачів повітря, таких як тверді частки, сажа та інші, що дозволяє користувачеві оцінити вплив цих забруднень на здоров'я [11];

– засоби оцінювання ризиків для здоров'я: програмне забезпечення AirQ+ розраховує короткостроковий і довгостроковий вплив забруднювачів, враховуючи дані епідеміологічних досліджень. Це дозволяє оцінити ризик смертності та інші показники здоров'я у залежності від рівнів забруднень повітря [11];

– засоби підтримки експертів: AirQ+ рекомендується використовувати спільно з фахівцями з епідеміології та забруднення повітря, оскільки для правильної інтерпретації результатів важливо мати додатковий експертний контекст [11];

– розширені можливості аналізу: окрім стандартних методів, програма використовує різні інструменти, включаючи алгоритми рою частинок та штучні нейронні мережі для більш точної оцінки впливу забруднень повітря на здоров'я [11];

– засоби оцінювання різноманітності забруднень: програма враховує широкий спектр забруднювачів, не обмежуючись лише загальноживаними. Вона дозволяє враховувати токсичні речовини, які можуть бути менш поширеними, але мають суттєвий вплив на якість повітря [11];

– засоби створення комфортного середовища: технологія EPI Air покращує якість повітря, зменшуючи пил у приміщеннях, що веде до кращих умов проживання, а також зниження впливу статичної електрики [11].

Серед недоліків програмного забезпечення EPI Air можна відзначити такі [11]:

- обмежена застосовуваність: програмне забезпечення EPI Air можна використовувати для закритих систем, що не дозволяє його застосовувати у різних природних системах відкритого типу для аналізу якості повітря [11];

- обмежена точність: програмне забезпечення AirQ+ може мати обмежену точність при прогнозуванні впливу забруднень на здоров'я, оскільки залежить від доступної інформації та методів розрахунків [11];

- специфічні умови застосування: методи, які використовує програмне забезпечення AirQ+ для оцінки впливу, можуть бути специфічними для певних ситуацій, що ускладнює її універсальне використання [11];

- складність інтерпретації результатів: отримані результати можуть бути складними для інтерпретації для осіб без спеціалізованого фахового знання з епідеміології або екології [11].

Програмне забезпечення ADMS-Forecast [12] дозволяє виконувати надання прогнозів якості повітря в режимі онлайн з великою точністю. Ця система використовує локальну інформацію про тенденції руху, погодні прогнози та регіональні прогнози щодо складу повітря. Ці дані є частиною програмної системи моделювання розподілу забруднювачів у повітрі у місті, надаючи деталізовані концентрації з високою просторовою роздільною здатністю[12].

Ці прогнози доступні через кольорові карті забруднення, текстові та числові підсумки, мобільні додатки, Twitter та Facebook, а також через SMS-повідомлення, голосову пошту та електронну пошту [12].

Програмне забезпечення ADMS-Forecast (рис. 1.3) надає можливість прогнозування рівня забруднення повітря на кілька днів вперед, що дозволяє людям контролювати своє здоров'я. Особливо це корисно для пацієнтів з астмою, які можуть відчувати погіршення через забруднене повітря, особливо внаслідок фізичних навантажень чи перебування на відкритому повітрі [12].

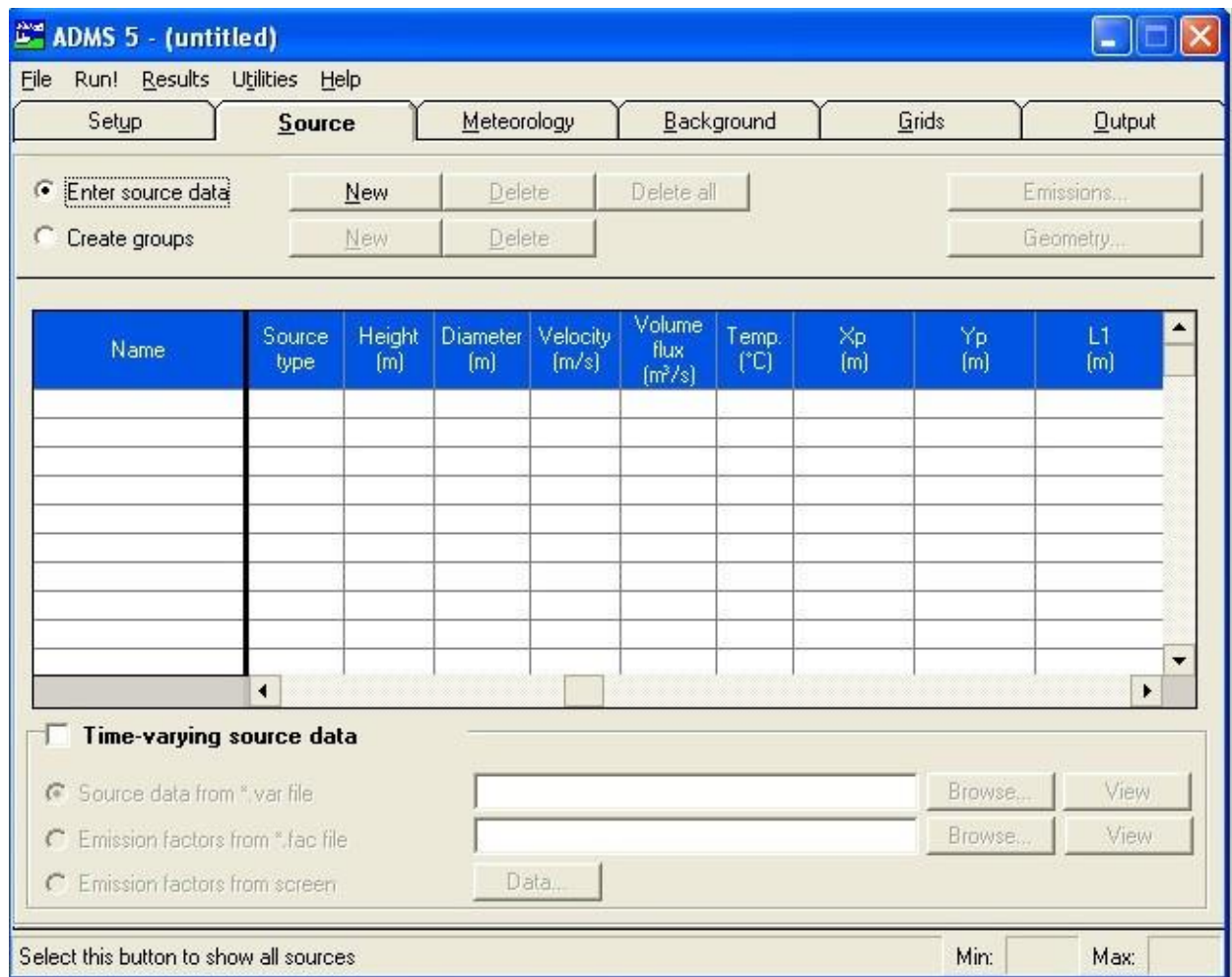


Рисунок 1.3 – Програмне забезпечення ADMS-Forecast [12]

Базові дані про використання забруднюючих речовин отримуються програмним забезпеченням ADMS-Forecast з національних реєстрів викидів та місцевих джерел. Моніторинг за допомогою програмного забезпечення ADMS-Forecast охоплює ключові забруднюючі речовини, такі як транспортні потоки на головних дорогах, виробництво, викиди з житлових і комерційних районів, а також загальне забруднення повітря [12].

Для прогнозування рівня забруднення повітря ці викиди враховуються у моделях, які враховують прогнозовані значення ключових метеорологічних показників: швидкість та напрямок вітру, температура та ступінь хмарності. Програмна система ADMS-Forecast використовує обмеження моделей для зон масштабного рівня та інтегрується з регіональними службами різних

населених пунктів. Сервіс надає безкоштовні європейські прогнози через певні програмні інтерфейси [12].

Програмне забезпечення ADMS-Urban є системою керування якістю повітря, розробленою для міського планування та досліджень якості повітря. Вона застосовується в більшості британських міст, а результати були значно підтверджені у наукових дослідженнях [12]. Її ефективність перевірена не лише у Великобританії, а й у різних куточках світу, де вирішуються питання, щодо впливу різних джерел забруднення (таких як промислові викиди, транспорт та опалення), а також урахування складності топографії, наприклад, розповсюдження забруднення. Дослідження з використання ADMS-Urban охоплюють розробку планів управління якістю повітря у різних містах світу; моделювання транспортних ресурсів у Каліфорнії, моделювання викидів від вугілля в Белфасті [12].

Однією з ключових переваг програмного забезпечення ADMS-Forecast є можливість створення високоякісних карт. Це можуть бути кольорові карти якості повітря для регіональних областей або різні мапи для конкретних вулиць, районів або міст. Ці карти можуть бути легко адаптовані для різних носіїв, включаючи публікації в Інтернеті [12].

Програмне забезпечення ADMS-Forecast також підтримує текстові повідомлення, голосову пошту та електронні листи щодо якості повітря, а також використання Twitter і Facebook для сповіщень. Користувачі можуть підписатися на отримання таких сповіщень в Інтернеті або, надіславши повідомлення на спеціальний короткий номер. Прогнози також доступні через смартфон або у вигляді PDF-бюлетенів [12].

Серед особливостей програмного забезпечення ADMS-Forecast можна виділити такі [12]:

- засоби прогнозування якості повітря: програмне забезпечення ADMS-Forecast надає можливість прогнозувати рівні забруднення повітря на кілька днів вперед з високою точністю, що дозволяє громадянам контролювати вплив на їхнє здоров'я [12];

– можливість моделювання викидів: програмне забезпечення ADMS-Forecast моделює викиди та поширення забруднюючих речовин, враховуючи ключові метеорологічні параметри, такі як швидкість та напрямок вітру, температура та хмарність [12];

– врахування регіональних особливостей: програмне забезпечення ADMS-Forecast забезпечує зв'язок з регіональними службами для різних населених пунктів, що дозволяє інтегрувати місцеву інформацію про прогнози погоди та якість повітря [12];

– система сповіщень: програмне забезпечення ADMS-Forecast надає можливість користувачам отримувати сповіщення про якість повітря через SMS, голосову пошту, соціальні мережі та електронну пошту, що дозволяє широко впроваджувати інформацію для громадськості [12];

– засоби доступу: програмне забезпечення ADMS-Forecast забезпечує можливість доступу до прогнозів через вебпублікації, смартфони або інші програми для більшої доступності та зручності користувачів [12];

– засоби аналізу якості повітря: програмне забезпечення ADMS-Forecast забезпечує оцінку рівнів різних забруднювачів повітря, таких як тверді частки, сажа та інші, що дозволяє користувачеві оцінити вплив цих забруднень на здоров'я [12];

– засоби оцінювання ризиків для здоров'я: програмне забезпечення ADMS-Forecast розраховує короткостроковий і довгостроковий вплив забруднювачів на здоров'я людей, враховуючи дані епідеміологічних досліджень. Це дозволяє оцінити ризик смертності та інші показники здоров'я у залежності від рівнів забруднень повітря [12];

– засоби підтримки експертів: ADMS-Forecast рекомендується використовувати спільно з фахівцями з епідеміології та забруднення повітря, оскільки для правильної інтерпретації результатів важливо мати додатковий експертний контекст [12];

– розширені можливості аналізу: окрім стандартних методів, програма використовує різні інструменти, включаючи алгоритми рою частинок та

штучні нейронні мережі для більш точної оцінки впливу забруднень повітря на здоров'я [12];

- засоби оцінювання різноманітності забруднень: програма враховує широкий спектр забруднювачів, не обмежуючись лише загальновживаними. Вона дозволяє враховувати токсичні речовини, які можуть бути менш поширеними, але мають суттєвий вплив на якість повітря [12];

- динамічність досліджень: програмне забезпечення ADMS-Forecast орієнтовано на актуальність даних і дозволяє проводити динамічні аналізи змін впливу забруднювачів на здоров'я населення в залежності від розвитку ситуації з якістю повітря [12];

- універсальність застосування: ADMS-Forecast можна використовувати для різних цілей: від оцінки загального впливу забруднень повітря на населення до аналізу конкретних забруднювачів та їх впливу на здоров'я [12].

Серед недоліків ADMS-Forecast можна відзначити такі [12]:

- обмежена точність: програмне забезпечення ADMS-Forecast може мати обмежену точність при прогнозуванні впливу забруднень на здоров'я, оскільки залежить від доступної інформації та методів розрахунків [12];

- потреба в даних: робота програми програмного забезпечення ADMS-Forecast вимагає наявності відповідних даних про забруднення повітря, які можуть бути обмеженими або відсутніми для деяких регіонів чи періодів [12];

- специфічні умови застосування: методи, які використовує програмне забезпечення ADMS-Forecast для оцінки впливу, можуть бути специфічними для певних ситуацій, що ускладнює її універсальне використання [12];

- складність інтерпретації результатів: отримані результати можуть бути складними для інтерпретації для осіб без спеціалізованого фахового знання з епідеміології або екології [12].

Порівняльну характеристику програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей

Критерій порівняння	AirQ+	EPI Air	ADMS-Forecast
Засоби оцінювання ризиків для здоров'я	+	+—	+
Зручний посібник користувача	+	+—	+
Засоби аналізу якості повітря	+	+—	+
Розширені можливості аналізу	+	—	+
Універсальність застосування	+	—	+
Система оповіщень	—	+—	+
Підтримка експертів	+	—	+—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для оцінювання впливу шкідливих речовин на здоров'я людей. Проте деякі програми не завжди можуть забезпечувати точні та достовірні результати оцінювання впливу шкідливих речовин на здоров'я людей, особливо у випадках, коли мережа датчиків недостатня або у вимірюванні домінує певна погрішність. Крім того, програмне та апаратне забезпечення не завжди може охопити усі області або забезпечити однакову точність результатів в усіх регіонах, що ускладнює загальне оцінювання стану повітря. Деякі програмні засоби можуть бути високо вартісними, особливо якщо потрібно розширення функціональності або доступ до більш деталізованих даних. Тому актуальною є розробка програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

При розробці програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей необхідно забезпечити такі функціональні вимоги:

- можливість прогнозування рівня захворюваності в залежності від показників забруднення повітря;
- використання нейромережевих технологій як базису для створення прогностичних моделей;
- врахування різних показників забруднення повітря, зокрема, обсягу викидів, а також інших показників (кількість населення, кількість лікарів, кількість опадів та ін.);
- можливість оцінювання рівня захворюваності на різні хвороби, зокрема, туберкульоз, захворюваності системи кровообігу та ін.;
- можливість виведення результатів як в текстовому, так і в графічному вигляді.

1.3 Висновки за розділом 1

Відзначено, що у наш час шкідливі речовини, які можна знайти у повітрі, охоплюють широкий спектр забруднюючих речовин. Проаналізовано фактори, які пов'язані з рівнем забруднення повітря та можуть впливати на рівні захворюваності населення. Підкреслено, що використання програмних засобів для оцінювання впливу шкідливих речовин в повітрі на здоров'я людей залишається вкрай актуальним. Завдяки таким засобам можна аналізувати рівень забруднення повітря, визначати концентрацію токсичних речовин та їх вплив на здоров'я. Це дозволяє урядовим установам, організаціям охорони здоров'я та громадськості реагувати на шкідливі викиди, впроваджувати заходи для зменшення забруднення та забезпечувати захист громадського здоров'я. Таке програмне забезпечення створює можливість отримання точних та достовірних даних про якість повітря та сприяє управлінню екологічною ситуацією для покращення якості життя населення. Програмні засоби для оцінювання впливу шкідливих речовин в повітрі є важливим інструментом для визначення рівня забруднення атмосфери та його впливу на здоров'я людей. Ці програмні рішення зазвичай використовуються

для моніторингу та аналізу різноманітних параметрів повітря, таких як концентрація токсичних речовин, рівень забруднення, та інші параметри, які можуть мати негативний вплив на здоров'я. Вони допомагають у визначенні якості повітря, можливих джерел забруднення, відслідковують тенденції змін у рівнях різних речовин, а також надають аналіз даних, який сприяє прийняттю рішень у зв'язку з мінімізацією шкідливого впливу на довкілля та здоров'я людей.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для генерації текстової інформації. Проте деякі програми для генерації тексту можуть створювати стереотипні або загальні висловлювання, що робить їх менш корисними для унікальних або специфічних завдань, деякі програми можуть іноді генерувати неправильну або недостовірну інформацію, що може призвести до поширення некоректної інформації. Інші подібні програми для досягнення оптимальних результатів та уникнення неправильних висновків програми часто потребують постійного нагляду та втручання від фахівців для покращення генеративної моделі. Тому актуальною є розробка програмного забезпечення для генерації текстової інформації.

Сформульовано функціональні вимоги до програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей було використано мову програмування Python [13], [14].

Python – це універсальна мова програмування, яка спрямована на полегшення процесу розробки та зрозумілості коду [13], [14]. У Python відсутні обмеження для унікальності його синтаксису. Вбудована бібліотека містить різноманітні корисні функції. Мова підтримує різні підходи до програмування, зокрема структурний, об'єктно-орієнтований, функціональний та імперативний [13], [14]. Основні особливості включають динамічні типи даних, автоматичне управління пам'яттю, розширену перевірку винятків, обробку винятків, підтримку паралельних операцій і просту високорівневу структуру даних [13], [14].

Python організований за допомогою функцій та класів, які об'єднуються у модулі, а ці модулі, у свою чергу, можуть бути об'єднані в пакети [13], [14]. Ця мова програмування має кілька інсталяцій, а основний інтерпретатор – CPython, розроблений для роботи на різних платформах. Він розповсюджується за ліцензією Free Python Software Foundation і може використовуватися у різних програмах. Окрім CPython, є різні інші реалізації, що підтримують перенесення. Проект Рірі спрощує використання Python, зменшуючи витрати на зміну мови та тестування нових можливостей [13], [14]. Python постійно розвивається, з новими версіями, що виходять кожні два з половиною роки. Відсутність офіційного стандарту, такого як ANSI, ISO інших, разом із постійними оновленнями, робить CPython відомим серед інших реалізацій. Python підтримує динамічну типізацію, тобто типи змінних визначаються лише під час виконання програми [13], [14].

Вбудовані типи у мові програмування Python включають багато різних категорій, таких як логічні значення, символічні рядки, рядки Unicode, цілі

числа будь-якої точності, числа з плаваючою комою, комплексні числа та інше. Також є вбудовані структури даних, такі як списки, кортежі, словники та інші [13], [14].

У Python все – це об'єкт, навіть функції, методи, модулі та класи. Крім вбудованих типів, можна створювати свої власні, визначаючи нові типи через написання класів або в модулях розширень (наприклад, у мові C) [13], [14].

Система класів Python підтримує успадкування (одинарне та багатoshарове) і метапрограмування. Більшість вбудованих типів успадковують властивості від інших типів [13], [14].

У Python все можна розділити на атоми та сховища. Атоми – це основні типи даних, такі як цілі числа, довгі, складні та інші. Коли ви визначаєте атомарні об'єкти, їх значення копіюються, але вказівники на об'єкти копіюються лише як посилання на об'єкти. Тому, якщо змінити еталонний об'єкт, це може вплинути на всі посилання на нього, але не на всі доступні об'єкти. Наприклад, деякі об'єкти, такі як рядки та кортежі, є незмінними, тоді як інші, наприклад списки та словники, можна змінювати [13], [14].

Кортежі у Python схожі на незмінні списки. Зазвичай вони працюють швидше, ніж списки, особливо, якщо ви не плануєте змінювати їхній порядок елементів [14].

Python підтримує різні типи даних, такі як логічні значення, рядки, цілі числа будь-якої точності, числа з плаваючою комою, комплексні числа та інші. Також має вбудовані колекції, такі як списки, кортежі, словники та інші структури даних. У Python все є об'єктом, навіть функції, методи, модулі та класи. Крім цього, можна створювати нові типи даних, визначаючи класи або використовуючи модулі розширень (наприклад, у мові C) [13], [14].

Отже Python є мовою програмування, що відома своєю простотою та читабельністю коду. Вона використовується у багатьох сферах, включаючи веброзробку, наукові дослідження, штучний інтелект, обробку даних та багато іншого. Одна з його переваг – широкий вибір бібліотек і фреймворків, які сприяють в розробці програм та розв'язанні різноманітних завдань. Python

також має активну спільноту, що постійно розвивається, що сприяє поширенню нових ідей та інновацій у світі програмування [13], [14].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей

Критерій порівняння мов програмування	Мова програмування		
	Python	Matlab	C++
Зручність розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей	+	—	—
Робота з числовими та іншими типами даних	+	+—	+
Функції успадкування	+	+—	+
Засоби створення нейромережевих моделей	+	+	+—
Наявність наборів інструментів для створення інтелектуальних систем	+	+	+—

Отже, для реалізації програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей обрано мову програмування Python, орієнтована на загальні завдання у програмуванні та спрямована на поліпшення продуктивності розробника та зрозумілість коду. Крім того, ця

мова дозволяє ефективно працювати з засобами штучного інтелекту та створювати високо точні моделі для класифікації та прогнозування.

2.2 Вибір середовища розробки для розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей

В якості середовища розробки для створення програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей було обрано PyCharm [15], [16].

PyCharm, розроблене компанією JetBrains, є інтегрованим середовищем для роботи з мовою програмування Python. Це програмне забезпечення пропонує ряд корисних інструментів, включаючи аналіз коду, можливості візуальної налагодження та тестування модулів, і також підтримує веб-розробку на Django та Flask. PyCharm доступний для операційних систем Windows, Mac OS X і Linux. Цей інструмент не лише допомагає в аналізі та оптимізації коду, але і забезпечує ефективне виправлення помилок та тестування модулів, що сприяє підвищенню якості розробки програмного забезпечення [15], [16].

Середовище розробки PyCharm підтримує веброзробку через популярні фреймворки, такі як Django та Flask, що робить його невід'ємним інструментом для фахівців у сфері розробки веб-проектів. Це інтегроване середовище розробки працює на платформах Windows, Mac OS X і Linux, забезпечуючи універсальність та зручність для розробників, які працюють на різних операційних системах. PyCharm не лише інструмент – це надійний помічник у створенні та вдосконаленні програмного забезпечення, який надає широкий набір засобів та функцій для роботи з Python [15], [16].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей

Критерій порівняння середовищ розробки	Середовище розробки		
	PyCharm	Visual Studio	Sublime
Можливості розробки за допомогою Python	+	+—	+
Інструменти для аналізу програмного коду	+	+	+—
Можливості візуальної розробки	+	+	—
Зручність для створення ПЗ для оцінювання впливу шкідливих речовин на здоров'я людей	+	—	—
Зручна орієнтація у проєкті та вихідному коді програми	+	—	—

Основні функції PyCharm включають [15], [16]:

- комплексний аналіз коду та виявлення синтаксичних помилок;
- відображення вихідного коду програмного забезпечення для зручної навігації по проєкту та швидкого перемикавання між структурами, файлами, класами та методами [15], [16];
- переформатування функцій, перейменування, видалення, додавання змінних, методів і констант [15], [16];
- інструменти веброботи, що підтримують фреймворк розробки Django [15], [16];
- вбудований засіб налагодження для Python [15], [16];
- інтеграція ресурсів модульного тестування [15], [16];

- підтримка Google App Engine [15], [16];
- робота з різними системами контролю версій, такими як Mercurial, Git, Subversion, Perforce і CVS, із використанням одного і того ж програмного інтерфейсу [15], [16].

Таким чином, для створення програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей обрано середовище розробки PyCharm, що є потужним інструментом для створення програм на Python, включаючи основні компоненти для роботи з засобами штучного інтелекту, які необхідні для розробки програм, що оцінюють та прогнозують рівень захворюваності населення на основі даних про забруднення повітря.

2.3 Методи та моделі машинного навчання для оцінювання впливу шкідливих речовин

Для аналізу впливу викидів забруднюючих речовин на рівень захворюваності населення від таких викидів, важливим є використання методів штучного інтелекту та машинного навчання, які дозволяють виявити та узагальнити ці взаємозв'язки для майбутнього використання на практиці та створення відповідного програмного забезпечення [17]–[19].

Проаналізуємо методи та моделі штучного інтелекту, які можна використовувати для створення програмного забезпечення для оцінювання впливу шкідливих речовин.

Логістична регресія – це модель, яка оцінює ймовірність певної події, базуючись на значеннях певних характеристик. У цій моделі є одна залежна змінна, що має два стани – 0 (подія не відбувається) або 1 (подія відбувається), і є набір незалежних змінних, які використовуються для обчислення ймовірності значення залежної змінної [17]–[19].

Методи опорних векторів – це набір алгоритмів, які вирішують завдання класифікації та регресії. Основна ідея полягає у перенесенні вихідних векторів у простір більшої вимірності, після чого знаходиться

дискретна гіперплощина з найбільшим відхиленням у цьому просторі. По обидві сторони цієї гіперплощини визначаються ще дві гіперплощини, що створюють розділову гіперплощину, максимізуючи відстань між двома паралельними гіперплощинами [17]–[19].

Метод найменших квадратів використовується для зменшення відхилень функції від суми змінних, що потребують визначення. У регресійному аналізі він оцінює невідомі параметри у регресійній моделі на основі системних вибірових даних. Мета полягає в тому, щоб знайти значення x , які найближче до заданого значення y для функцій [17]–[19].

Інший підхід до цієї проблеми полягає у використанні моделі на основі випадкового лісу або лісу випадкових дерев рішень. Це аналіз комітету осіб, які приймають рішення для вирішення проблем класифікації, регресії та інших, базуючись на створенні набору дерев рішень та рішенні щодо відмінностей між різними класами (для класифікації) або прогнозуванні окремих дерев (для регресії) [17]–[19].

Метод ближнього сусіда ґрунтується на припущенні, що об'єкт, який ми класифікуємо, подібний до об'єктів тестової вибірки. Це також відоме як алгоритм k -найближчих сусідів. Він знаходить відомі об'єкти серед усіх знайдених, які схожі на раніше невідомі об'єкти. Рішення приймаються з урахуванням суміжності близьких сусідів до певних класів. Важливо вибрати коефіцієнт k , що визначає кількість сусідів [17]–[19].

Після вибору k ми знаходимо k найближчих сусідів – об'єкти, найбільш близькі до вектора ознак нового об'єкта. Потім приймається рішення щодо їх впливу на клас нових об'єктів, використовуючи комбіновану функцію, яка може мати різні варіанти. Один з них, незважене голосування, гарантує, що обрані k найближчих сусідів мають однакові голоси та ранги, вважаючи, що вони проголосували за один і той же клас. Таким чином, новий об'єкт призначається до класу з найбільшою кількістю голосів [17]–[19].

У випадку зваженого голосування враховується відстань між об'єктом голосування та новим об'єктом. Остаточне рішення щодо класу нового об'єкта здійснюється на основі голосів та врахування відстані між об'єктами [17]–[19].

Штучні нейронні мережі відбивають структуру та роботу біологічних нейронних систем у тварин. Вони складаються з штучних нейронів, які імітують функціонування природних нейронів та взаємодію між ними. Ці мережі базуються на математичних моделях і працюють разом, щоб обробляти інформацію, подібно до того, як працює справжній, біологічний, мозок. Кожен штучний нейрон отримує та відправляє сигнали і може змінювати свій стан в залежності від отриманих сигналів, що допомагає вирішувати конкретні завдання [17]–[19]. Математично, штучний нейрон подібний до вагового склоочисника: його вихідний сигнал залежить від ваги синапсів та вхідних сигналів [17]–[19]. Одна з ключових властивостей нейромережових моделей – можливість навчання. У біологічних системах навчання передбачає коригування синаптичних зв'язків між нейронами. Це також застосовується до штучних нейронних мереж. Математично, цей процес полягає у встановленні коефіцієнтів зв'язку між штучними нейронами в мережі [17]–[19].

Існують два способи навчання: навчання з вчителем та без нього. У першому випадку мережа отримує вхідні та вихідні дані, відповідні цим вхідним даним. Вона порівнює отримані результати з бажаними виходами і коригує синаптичні зв'язки відповідно до помилок. Цей процес повторюється кілька разів під час навчання на одних і тих самих даних [17]–[19].

Інший тип – навчання без вчителя. У цьому випадку система отримує лише вхідні дані без бажаного виходу. Під час навчання відбувається самоорганізація або системна адаптація [17]–[19].

Штучні нейронні мережі застосовуються для розв'язання різноманітних завдань, таких як класифікація зображень, кластеризація даних, прогнозування, оптимізація та інші. Проте вони не завжди дають точні результати, а їх якість оцінюється відношенням правильних та неправильних відповідей [17]–[19].

Один з видів таких мереж – мережа з довгою короткостроковою пам'яттю (LSTM, Long Short-Memory Neural Network). У повторюваних нейронних мережах присутні цикли, які передають активації з попередніх кроків у мережу на поточний крок. Ці активації зберігаються у внутрішньому просторі мережі, що може зберігати тимчасову контекстну інформацію протягом тривалого часу. Цей метод дозволяє динамічно коригувати вікно історії вхідної послідовності [17]–[19]. Мережа з довгостроковою пам'яттю (рис. 2.1) використовує окремі вузли, призначені для тривалого зберігання інформації, інтегрованих у рекурентну нейронну мережу.

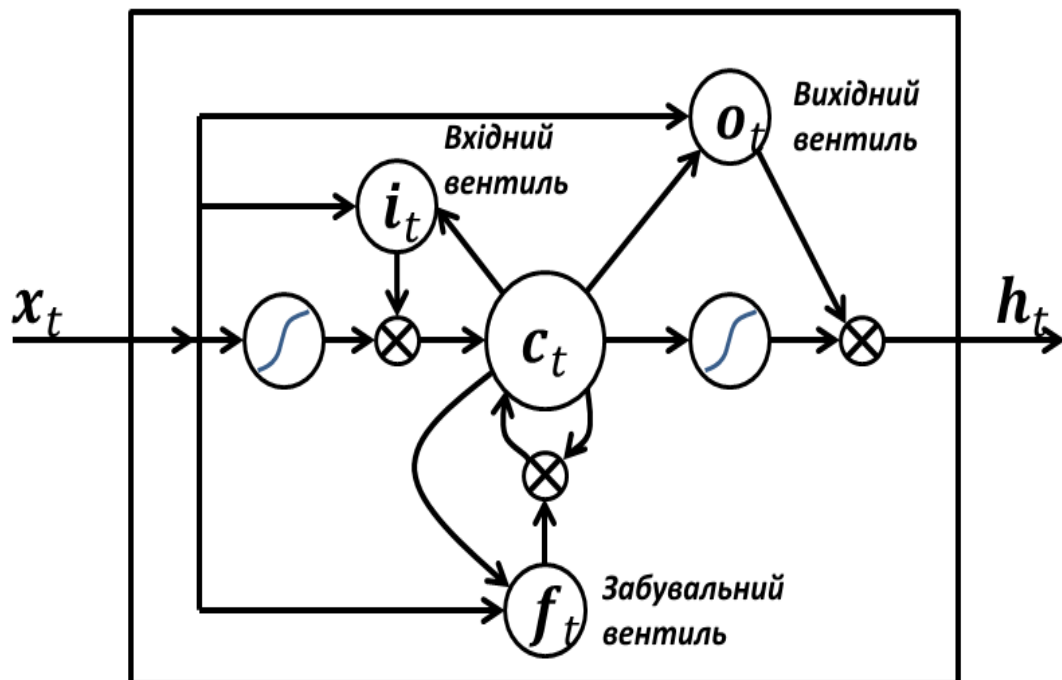


Рисунок 2.1 – Нейронна мережа з довгою короткостроковою пам'яттю

Ці вузли можуть зберігати значення протягом тривалого періоду без зміни завдяки виключенню їх від функції активації, що дозволяє зберігати дані без їх спотворення. Зазвичай вузли довгострокової пам'яті організовані в блоки, які містять кілька таких вузлів. Ця конструкція подібна до глибоких мереж з багатьма шарами, спрощуючи реалізацію за допомогою паралельних обчислень [17]–[19].

Блоки довгострокової пам'яті мають три або чотири "клапани", що регулюють потік інформації у пам'ять. Зазвичай ці клапани використовують логістичну функцію активації. Є три типи клапанів: "вхідний", "вихідний" та "забутий". "Вхідний" контролює вхідні сигнали у вузол, "вихідний" відповідає за обчислення сигналів у вузлі та його вихід, а "забутий" відповідає за збереження або видалення інформації у вузлі [17]–[19].

Обидва клапани в процесі навчання відкривають та закривають доступ до кожної комірки пам'яті. Якісне введення є важливим, оскільки це допомагає уникнути конфліктів, які можуть виникнути внаслідок надмірного введення даних при навчанні нейронної мережі. Аналогічно, якісний вихідний зв'язок захищає інші пристрої від можливих перешкод, спричинених непотрібним вмістом тимчасової пам'яті [17]–[19].

На сьогоднішній день практично всі популярні програмні пакети прикладних програм включають пакети плагінів для моделювання штучних нейронних мереж з високорівневим API [17]–[19].

Еволюційні алгоритми можуть бути застосовані для розв'язання завдань оптимізації або пошуку за певних умов. Спочатку, проблему необхідно відобразити у вигляді послідовності, правильно визначивши оператори, переходи та перетини цих послідовностей, а також встановити функції, що оцінюють результати таких дій [17]–[19].

Оператор кросинговеру дозволяє внести значні зміни у послідовність та проводить пошук серед можливих варіантів розв'язків, тоді як оператор мутації «адаптується» до найкращого розв'язку [17]–[19]. Генетичні алгоритми використовуються для вирішення завдань, які можна відобразити у вигляді різних значень, подібних до генів у хромосомі. Спочатку генерується випадковий набір початкових "особин" — початкова популяція. Кожна особина оцінюється за допомогою певної функції, яка відображає її пристосованість. Після оцінки для кожної особини визначаються показники її успішності [17]–[19].

Далі, враховуючи ці значення, обираються припустимі перетини. Особини, які пройшли через перетини, піддаються спеціальним операторам, таким як кросингвер і мутація, які моделюють біологічні процеси. Через кросингвер відбувається створення нового покоління особин [17]–[19]. Цей цикл повторюється кілька разів, поки не будуть виконані умови завершення, такі як певна кількість ітерацій чи значення функції придатності (рис. 2.2).

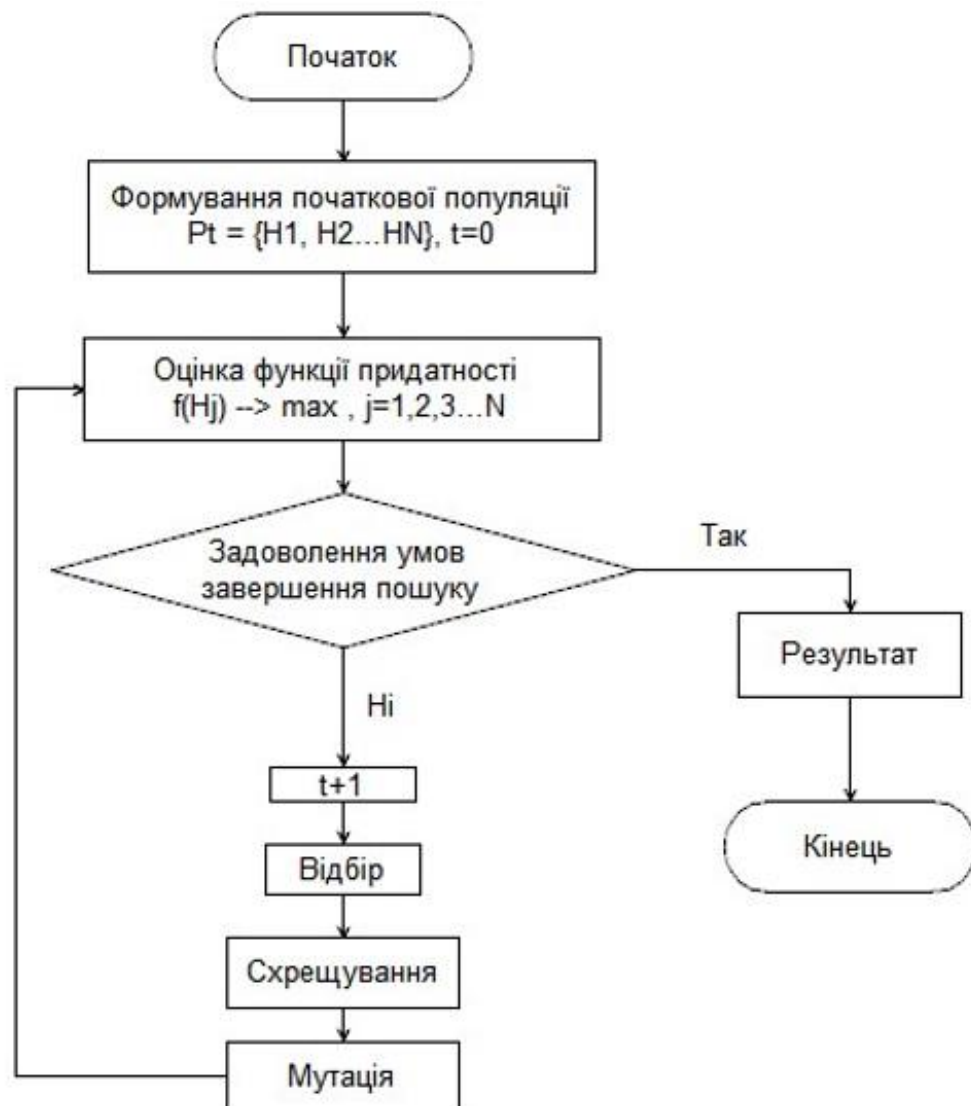


Рисунок 2.2 – Схема функціонування генетичного алгоритму [17]–[19]

Варто відзначити, що використання генетичного алгоритму часто ускладнюється вибором функції оцінки. Наприклад, у реальних задачах оптимізації структури цей процес може забирати багато часу. У таких

випадках може бути корисним побудова наближеної моделі та використання наближеного підходу [17]–[19]. Крім того, у деяких ситуаціях можливо виявити локальний, а не глобальний оптимум, що може привести до неправильного вирішення задачі [17]–[19].

Таким чином, на основі проведеного аналізу можна зробити висновок, що для розробки програмного забезпечення для оцінювання впливу шкідливих речовин необхідно створити модель, що надасть можливість оцінювати вплив шкідливих речовин на рівень захворюваності населення. Як базис для створення таких моделей доцільно обрати штучні нейронні мережі, які дозволяють з високою точністю прогнозувати значення вихідного параметру за заданими наборами вхідних ознак.

2.4 Розробка моделі програмного забезпечення для оцінювання впливу шкідливих речовин

З метою можливості розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей створимо відповідну інформаційну модель.

При розробленні моделі програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей враховуємо те, що користувачі програми мають можливості прогнозування рівня захворюваності в залежності від показників забруднення повітря, використання нейромережевих технологій як базису для створення прогностичних моделей, врахування різних показників забруднення повітря, зокрема, обсягу викидів, а також інших показників (кількість населення, кількість лікарів, кількість опадів та ін.), можливість оцінювання рівня захворюваності на різні хвороби, зокрема, туберкульоз, захворюваності системи кровообігу та ін., а також виведення результатів як в текстовому, так і в графічному вигляді.

Враховуючи особливості вимог до програмного забезпечення для оцінювання впливу шкідливих речовин, модель розробки програмного

забезпечення для оцінювання впливу шкідливих речовин, може бути подана у вигляді, зображеному на рис. 2.3.

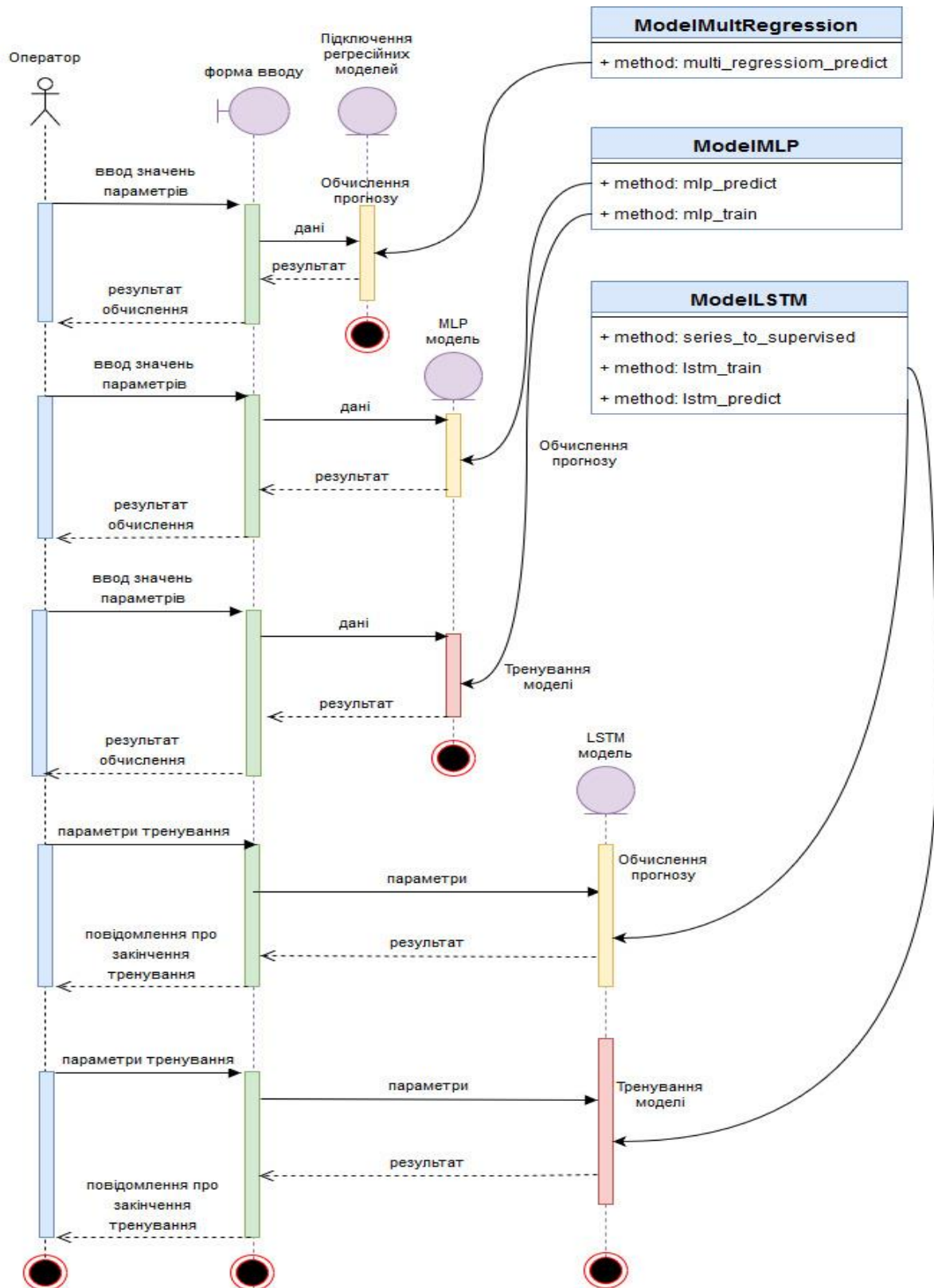


Рисунок 2.3 – Модель програмного забезпечення для оцінювання впливу шкідливих речовин

Таким чином, створено модель програмного забезпечення для оцінювання впливу шкідливих речовин, яка подана у вигляді діаграми та дозволяє розроблювати програмне забезпечення для оцінювання впливу шкідливих речовин.

2.5 Висновки за розділом 2

Для реалізації програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей обрано мову програмування Python, яка орієнтована на загальні завдання у програмуванні та спрямована на поліпшення продуктивності розробника та зрозумілість коду. Крім того, ця мова дозволяє ефективно працювати з засобами штучного інтелекту та створювати високо точні моделі для класифікації та прогнозування.

Для створення програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей обрано середовище розробки PyCharm, що є потужним інструментом для створення програм на Python, включаючи основні компоненти для роботи з засобами штучного інтелекту, які необхідні для розробки програм, що оцінюють та прогнозують рівень захворюваності населення на основі даних про забруднення повітря.

На основі проведеного аналізу зроблено висновок про те, що для розробки програмного забезпечення для оцінювання впливу шкідливих речовин необхідно створити модель, що надасть можливість оцінювати вплив шкідливих речовин на рівень захворюваності населення. Як базис для створення таких моделей доцільно обрати штучні нейронні мережі, які дозволяють з високою точністю прогнозувати значення вихідного параметру за заданими наборами вхідних ознак.

Створено модель програмного забезпечення для оцінювання впливу шкідливих речовин, яка подана у вигляді діаграми та дозволяє розроблювати програмне забезпечення для оцінювання впливу шкідливих речовин.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для оцінювання впливу шкідливих речовин

Структуру програмного забезпечення для оцінювання впливу шкідливих речовин подано на рис. 3.1.

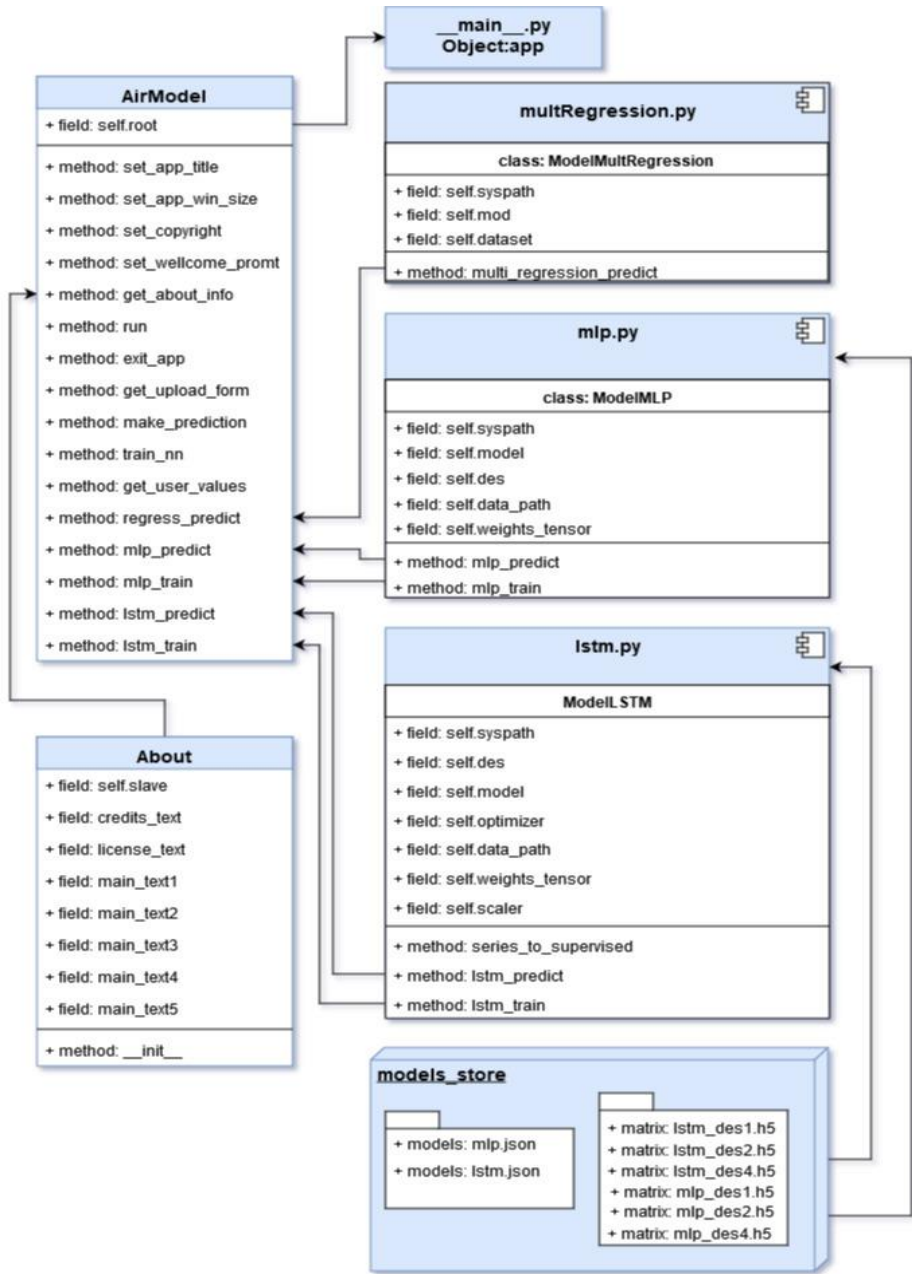


Рисунок 3.1 – Структура програмного забезпечення для оцінювання впливу шкідливих речовин

Розробка програмного забезпечення для оцінювання впливу шкідливих речовин охоплює створення набору програмних модулів, які дозволяють моделювати вплив концентрації забруднювачів повітря на рівень захворюваності. Цей набір включає модулі для прогнозування та розв’язання пов’язаних проблем, а також можливість використання створених прогностичних моделей.

Проаналізуємо основні розроблені програмні модулі:

- файл `main.py` є основним файлом проекту, де ініціалізується програма та створюється об’єкт головного класу `AirModel` програми, який запускає головний цикл роботи програми. В класі `AirModel` реалізовані методи, наведені нижче;

- `__init__` – конструктор класу `AirModel`. У цьому методі ініціалізуються змінні класу та елементи інтерфейсу;

- `set_app_title` – внутрішній метод, призначений для задання заголовка головного вікна програми;

- `set_app_win_size` – метод, що встановлює максимальний розмір вікна програми;

- `set_copyright` – функція, яка форматує відображення елемента з описом прав у головному вікні програми;

- `set_welcome_prompt` – процедура, що виводить відповідне повідомлення після успішного запуску програми;

- `get_about_info` – метод, що створює окреме вікно з інформацією про програму;

- `run` – функція, яка розпочинає основний цикл виконання програми;

- `exit_app` – метод, який завершує роботу програми;

- `get_upload_form` – процедура, що перевіряє, чи навчальний набір даних завантажено у програму;

- `make_prediction` – контроль процесу прогнозування;

- `train_nn` – метод, що реалізує модель контролера навчання;

- `get_user_values` – перевірка даних, введених користувачем;

– `regres_predict` – створення екземпляра класу `ModelMultRegression` та проведення прогнозних розрахунків з використанням моделей на основі логістичної регресії, опорного вектора, методів найменших квадратів, випадкового лісу та методів найближчих сусідів;

– `mlp_predict` – створення екземпляру класу `ModelMLP` та виконання розрахунків прогнозу за допомогою багатошарової моделі на основі перцептрона;

– `lstm_predict` – створення екземпляру класу `ModelLSTM` та виконання прогнозних розрахунків за допомогою нейронної моделі на основі мережі довгострокової пам'яті.

– `mlp_train` – створення екземпляру класу `ModelMLP` (модуль `mlp.py`) та навчання моделі за допомогою багатошарового перцептрона;

– `lstm_train` – створення екземпляру класу `ModelLSTM` (модуль `lstm.py`) та навчання моделі через мережу довгострокової пам'яті.

Модуль `MultRegression.py` містить клас, що здійснює розрахунок прогнозів за допомогою моделей, що базуються на логістичній регресії, опорних векторах, методах найменших квадратів, випадковому лісі та найближчому сусідів. Метод класу визначає прогнози рівня захворюваності.

Модуль `MLP.py` включає клас `ModelMLP`, який використовується для багатошарової моделі на основі перцептрона. У ньому містяться два методи: `mlp_predict` і `mlp_train`, що використовуються в прогнозних розрахунках і процесі навчання моделі.

Файл `lstm.py` містить клас `ModelLSTM`, що включає три методи. Метод `series_to_supervised` формує набір даних зі статистичної вибірки, який використовується для навчання нейронної мережі з довгостроковою пам'яттю. Методи `lstm_predict` і `lstm_train` використовують модель мережі довгострокової пам'яті для здійснення прогнозних розрахунків та процесу навчання моделі.

Набір моделей (`model_store`) містить описи моделей, зокрема багатошарових перцептронів і мереж довгострокової пам'яті. Ці описи

представлені у форматі JSON (mlp.json і lstm.json), а також набори синаптичних ваг цих мереж. Вони використовуються для обчислень і діагностики прогнозів різних захворювань.

3.2 База даних програмного забезпечення для оцінювання впливу шкідливих речовин

Схему бази даних програмного забезпечення для оцінювання впливу шкідливих речовин наведено на рис. 3.2.

Усі набори даних, які використовуються у програмному забезпеченні для оцінювання впливу шкідливих речовин, зберігаються у форматі CSV з використанням системи керування базами даних MySQL та оптимізації для легшого зберігання та обробки початкових статистичних даних. Це дозволяє створювати такі набори даних з такою структурою (рис. 3.2) за допомогою бази даних ER-діаграми. Проаналізуємо основні таблиці бази даних програмного забезпечення для оцінювання впливу шкідливих речовин.

Таблиця "Країна" вміщує інформацію про країни, де відбувалося вимірювання. Таблиця "Регіон" (tab_area), де області відображаються за регіонами. Поля цієї таблиці пов'язані з іншими таблицями різними способами. Таблиця "Населення" містить дані про населення за областями та періодами вимірювань. Поля цієї таблиці також пов'язані з різними іншими таблицями.

У таблицю "Параметри забруднення" ведеться облік викидів забруднюючих речовин та вуглекислого газу від стаціонарних джерел;

Таблиця значення параметрів забруднення додає значення параметрів по регіонах і місяцях до таблиці "Показники забруднення". Таблиці "Підсектор" і "Ступінь забруднення" мають спільні зв'язки з багатьма параметрами у цих таблицях.

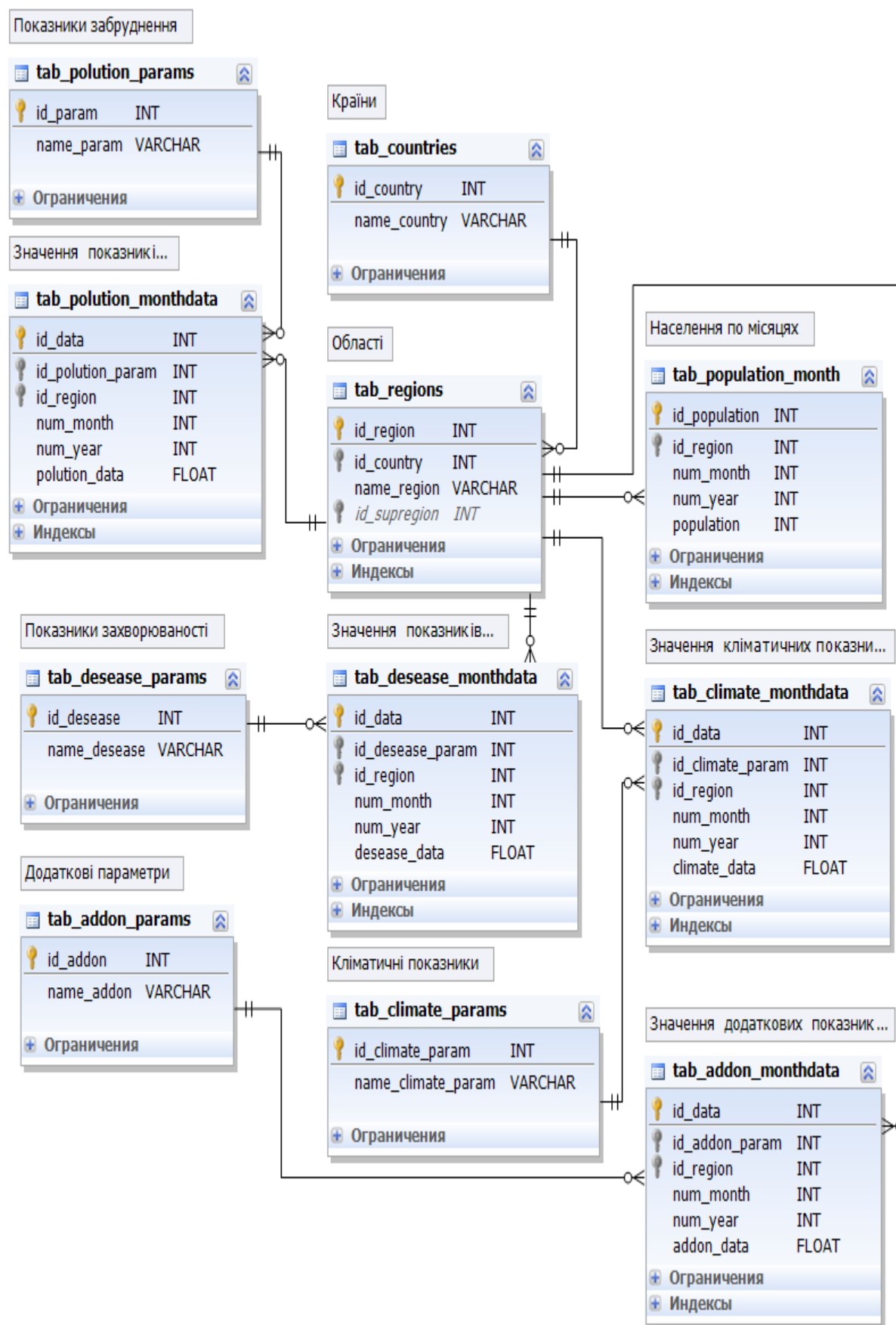


Рисунок 3.2 – Схема бази даних програмного забезпечення для оцінювання впливу шкідливих речовин

Таблиця "Кліматичні показники" містить дані про кліматичні умови. Таблиця "Значення кліматичних показників" містить значення кліматичних показників за регіонами і роками. Пов'язана з таблицями "регіон" і "індикатор клімату" через зв'язки багато до одного;

Таблиця "Симптоми захворювань" містить показники хвороб. Таблиця "Значення симптомів захворювань" включає значення захворюваності за областями і віком. Вона відображає відповідні показники захворюваності в співвідношенні багато до одного для різних областей.

Таблиця "Параметри програми" включає різні додаткові налаштування. Таблиця "Значення параметрів" містить додаткові значення параметрів, вона інтегрується з полями та розширеними таблицями параметрів і взаємодіє з ними.

3.3 Функціонування програмного забезпечення для оцінювання впливу шкідливих речовин

Схема функціонування програмного забезпечення для оцінювання впливу шкідливих речовин наведена на рис. 3.3.

Як видно з рис. 3.3 на першому етапі відбувається збирання інформації про рівні захворюваності населення на різні хвороби, а також відповідні дані про рівні забрудненості повітря (числові значення відповідних характеристик для кожного вимірювання). Потім ці дані використовуються для формування навчальної вибірки та заносяться до бази даних. На наступному етапі синтезується нейромережева модель для подальшого прогнозування рівнів захворюваності на певні хвороби в залежності від значень викидів різних забруднюючих речовин у повітря. Далі синтезована модель використовується для прогнозування рівнів захворюваності населення та виводяться результати прогнозування.

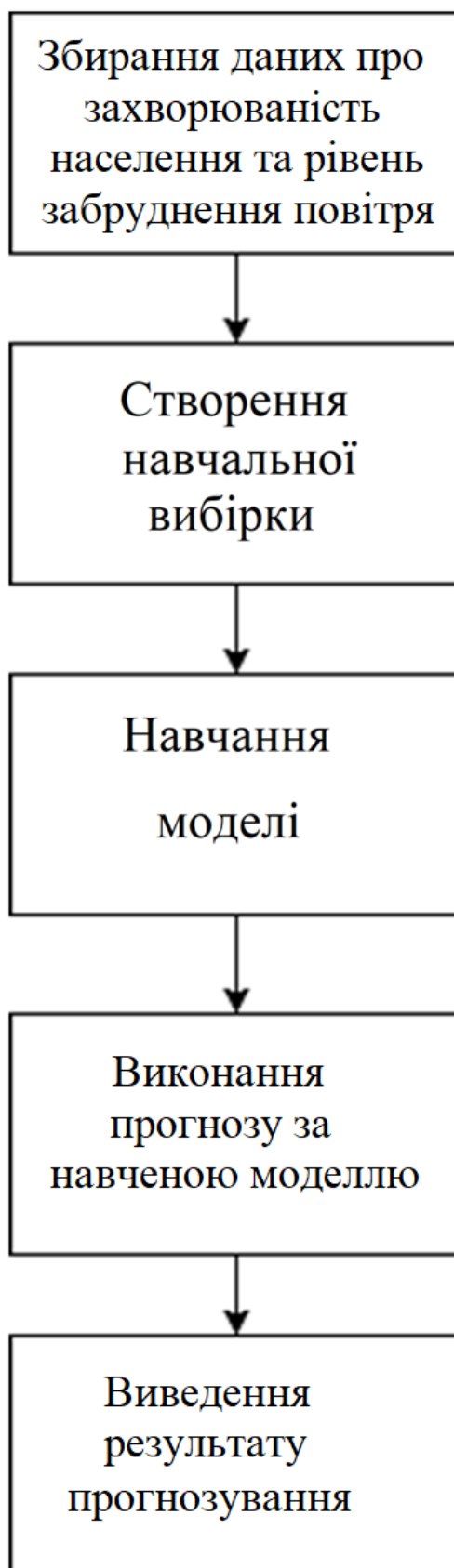


Рисунок 3.3 – Схема функціонування програмного забезпечення для оцінювання впливу шкідливих речовин

3.4 Проєктування інтерфейсу програмного забезпечення для оцінювання впливу шкідливих речовин

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для оцінювання впливу шкідливих речовин виконано з урахуванням вимог технічного завдання.

Згідно технічного завдання користувачі програми мають можливості прогнозування рівня захворюваності в залежності від показників забруднення повітря, використання нейромережевих технологій як базису для створення прогностичних моделей, врахування різних показників забруднення повітря, зокрема, обсягу викидів, а також інших показників (кількість населення, кількість лікарів, кількість опадів та ін.), можливість оцінювання рівня захворюваності на різні хвороби, зокрема, туберкульоз, захворюваності системи кровообігу та ін., а також виведення результатів як в текстовому, так і в графічному вигляді.

Головне вікно програмного забезпечення для оцінювання впливу шкідливих речовин наведено на рис. 3.4.

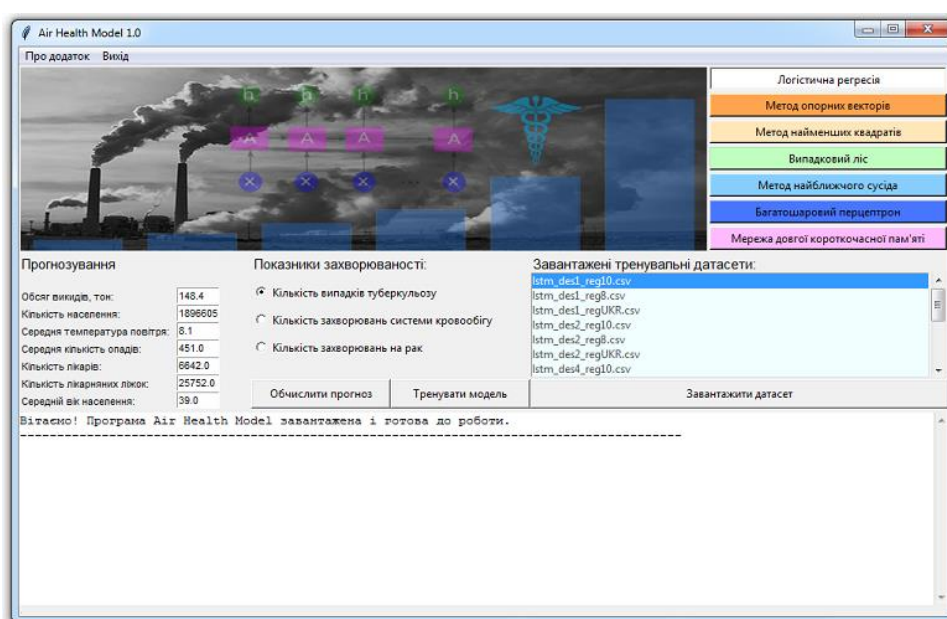


Рисунок 3.4 – Головне вікно програмного забезпечення для оцінювання впливу шкідливих речовин

Форма для введення показників захворюваності за попередній період наведена на рис. 3.5.

Прогнозування

Обсяг викидів, тон: 148.4
 Кількість населення: 1896605
 Середня температура повітря: 8.1
 Середня кількість опадів: 451.0
 Кількість лікарів: 6642.0
 Кількість лікарняних ліжок: 25752.0
 Середній вік населення: 39.0

Показники захворюваності:

☐ Кількість випадків туберкульозу
☒ Кількість захворювань системи кровообігу
☐ Кількість захворювань на рак

Завантажені тренувальні датасети:

lstm_des1_reg10.csv
 lstm_des1_reg10_month.csv
 lstm_des1_reg8.csv
 lstm_des1_reg8_month.csv
 lstm_des1_regUKR.csv
 lstm_des2_reg10.csv
 lstm_des2_reg8.csv

Вітаємо! Програма Air Health Model завантажена і готова до роботи.

Використана модель: Мережа довгої короточасної пам'яті
 Розрахунок для показника: Кількість захворювань системи кровообігу

Рисунок 3.5 – Форма для введення показників захворюваності за попередній період

Приклад форми з графіком результатів навчання синтезованої неймережевої моделі наведено на рис. 3.6.

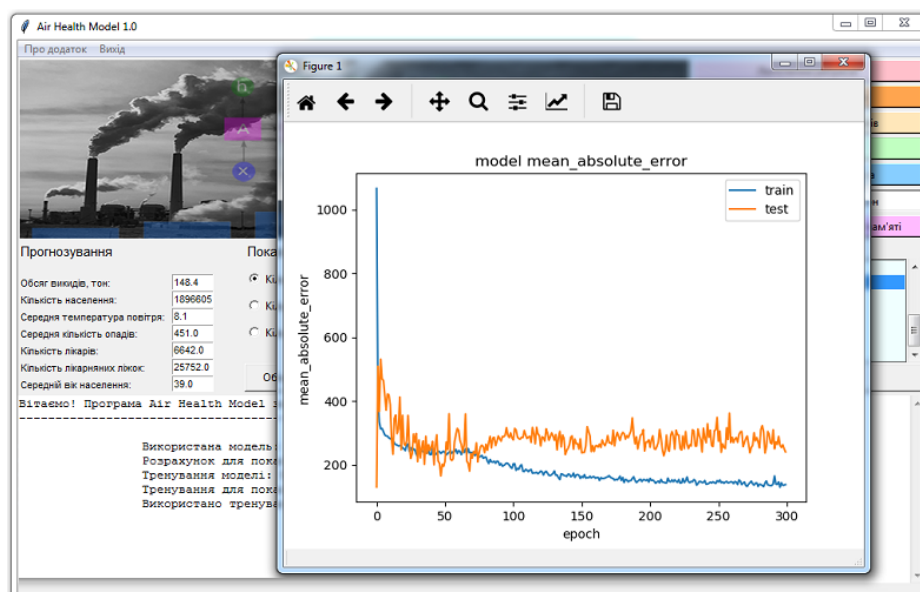


Рисунок 3.6 – Приклад форми з графіком результатів навчання синтезованої неймережевої моделі

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для оцінювання впливу шкідливих речовин. Запропоновано структуру та описано основні модулі програмного забезпечення для оцінювання впливу шкідливих речовин.

Описано функціонування програмного забезпечення для оцінювання впливу шкідливих речовин. Описано особливості реалізації програмного забезпечення для оцінювання впливу шкідливих речовин. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для оцінювання впливу шкідливих речовин.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для розв'язання задачі оцінювання впливу шкідливих речовин на рівень захворюваності населення. Програма написана на мові Python, орієнтована на загальні завдання у програмуванні та спрямована на поліпшення продуктивності розробника та зрозумілість коду. Крім того, ця мова дозволяє ефективно працювати з засобами штучного інтелекту та створювати високо точні моделі для класифікації та прогнозування. В якості середовища розробки для створення програми обрано PyCharm, що є потужним інструментом для створення програм на Python, включаючи основні компоненти для роботи з засобами штучного інтелекту, які необхідні для розробки програм, що оцінюють та прогнозують рівень захворюваності населення на основі даних про забруднення повітря.

Програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей забезпечує виконання таких функцій:

- можливість прогнозування рівня захворюваності в залежності від показників забруднення повітря;
- використання нейромережевих технологій як базису для створення прогностичних моделей;
- врахування різних показників забруднення повітря, зокрема, обсягу викидів, а також інших показників (кількість населення, кількість лікарів, кількість опадів та ін.);
- можливість оцінювання рівня захворюваності на різні хвороби, зокрема, туберкульоз, захворюваності системи кровообігу та ін.;
- можливість виведення результатів як в текстовому, так і в графічному вигляді.

Для експлуатації програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 1,5 Гб і більше;
- процесор із тактовою частотою 2,2 GHz і більше;
- вільне місце на диску 950 Мб або більше;
- встановлена операційна система.

Функціональні характеристики розробленого програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей наведено у додатку Б.

4.2 Характеристики програми для оцінювання впливу шкідливих речовин на здоров'я людей

Програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей може працювати як у консольному, так і візуальному режимах.

Програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей є набором бібліотек, що дозволяють моделювати та аналізувати вплив викидів забруднюючих речовин у повітря на показники захворюваності. Воно розроблене з урахуванням можливості використання інженерами-розробниками та легко адаптується для вирішення різних завдань. Також програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей включає інтерфейс, який дозволяє навіть не програмістам робити прогнози та навчати моделі на основі нейронних мереж, створюючи власну статистику.

Програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей є кросплатформним програмним забезпеченням, яке можна

використовувати на машинах під управлінням Windows і Unix. Програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей можна змінювати та поширювати згідно з ліцензією GPL 3, але має багато залежностей, які потребують уважного контролю, щоб уникнути можливих нестабільностей.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Рекомендується розгортати програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей у середовищі Python Anaconda.

Для запуску програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей необхідно запустити головний файл, що знаходиться в кореневій директорії програми.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей є модель, за якою будуть обчислюватися прогнозовані значення рівнів захворюваності, а також значення деяких характеристик забруднення повітря, зокрема, обсяг викидів, середня температура повітря, середній вік населення у регіоні, кількість населення та ін.

Вихідними даними програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей є синтезовані за набором вхідних даних моделі, що дозволяють оцінювати рівень захворюваності населення на різні хвороби за даними про рівні забруднення повітря; графіки, що відображають процес навчання прогностичної моделі; результати прогнозування за синтезованою моделлю.

4.3.3 Повідомлення

Користувач програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей може отримати такі повідомлення: повідомлення про необхідність обрання типу синтезованої моделі, повідомлення про необхідність введення вхідних даних та ін.

4.4 Виконання програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей

Після запуску програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей користувачу відображається головна форма (рис. 4.1).

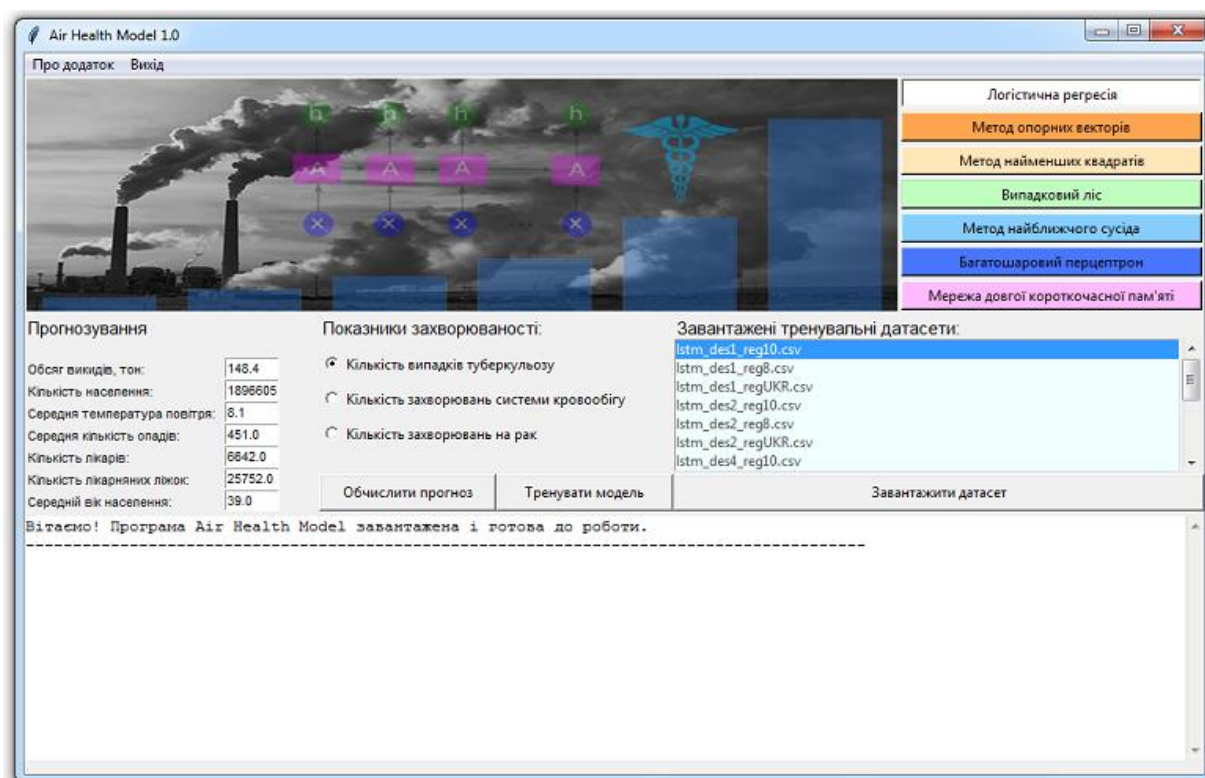


Рисунок 4.1 – Головне вікно програмного забезпечення для оцінювання впливу шкідливих речовин

Користувач може обирати прогностичні моделі, натискаючи відповідні кнопки у верхньому кутку справа головної форми.

Користувач має можливість тренувати нові моделі за наявними наборами даних. Для завантаження набору даних необхідно натиснути кнопку Завантажити датасет. Після цього необхідно натиснути кнопку Тренувати модель, внаслідок чого синтезується нова модель за заданим набором навчальних даних.

Важливо відзначити, що розробка моделі для навчання може бути витратною за ресурсами та зайняти певний час. Під час навчання інші функції програми будуть недоступні, поки обчислення не завершаться.

Після завершення навчання відобразиться ще одне вікно, де графічно будуть представлені результати навчання (динаміка помилок та матриця навчання мережі).

У нижній частині головної форми відображаються результати створення моделі та прогнозування за її допомогою.

При необхідності виконання прогнозів за допомогою синтезованих моделей користувач в області «Прогнозування» вводить значення, що характеризують поточну екологічну ситуацію, зокрема, кількість викидів, середню температуру повітря, а також інші параметри, наприклад, кількість та середній вік населення та ін. Далі користувач обирає, прогнозування рівня захворюваності якої хвороби необхідно зробити (туберкульоз, захворювання системи кровообігу та ін.). При натисканні на кнопку Обчислити прогноз виконується прогнозування за вже синтезованою моделлю.

Форма для введення показників захворюваності за попередній період наведена на рис. 4.2.

Після навчання моделі користувачу виводиться графік, що відображає процес її синтезу (значення помилки на навчальних та тестових даних), а також результати прогнозування за такою моделлю.

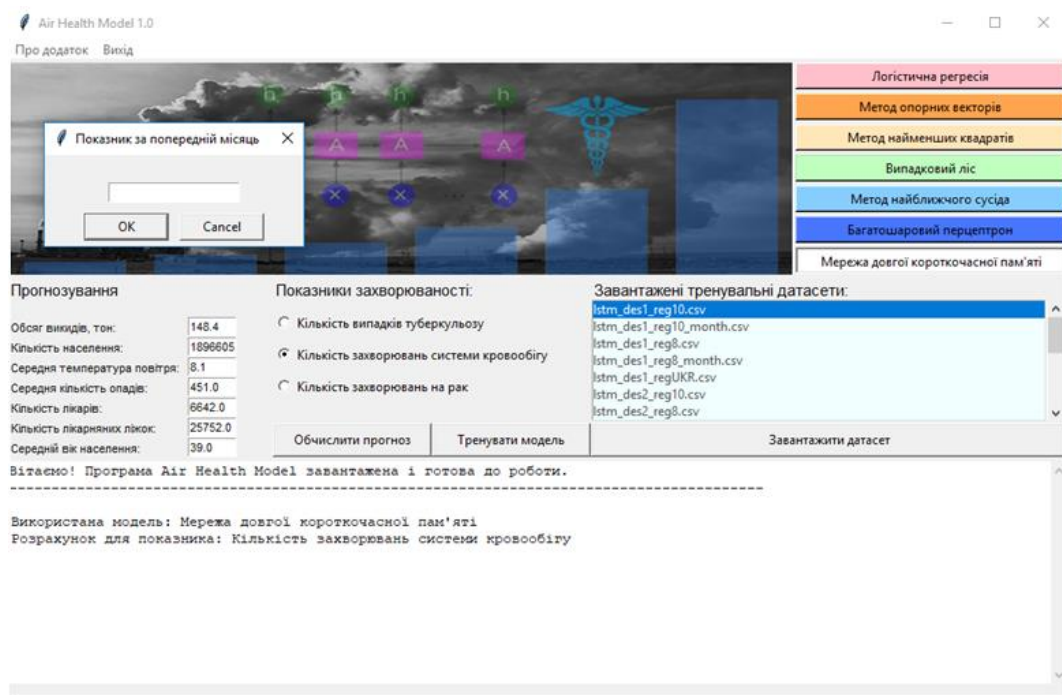


Рисунок 4.2 – Форма для введення показників захворюваності за попередній період

Приклад форми з графіком результатів навчання синтезованої нейромережевої моделі наведено на рис. 4.3.

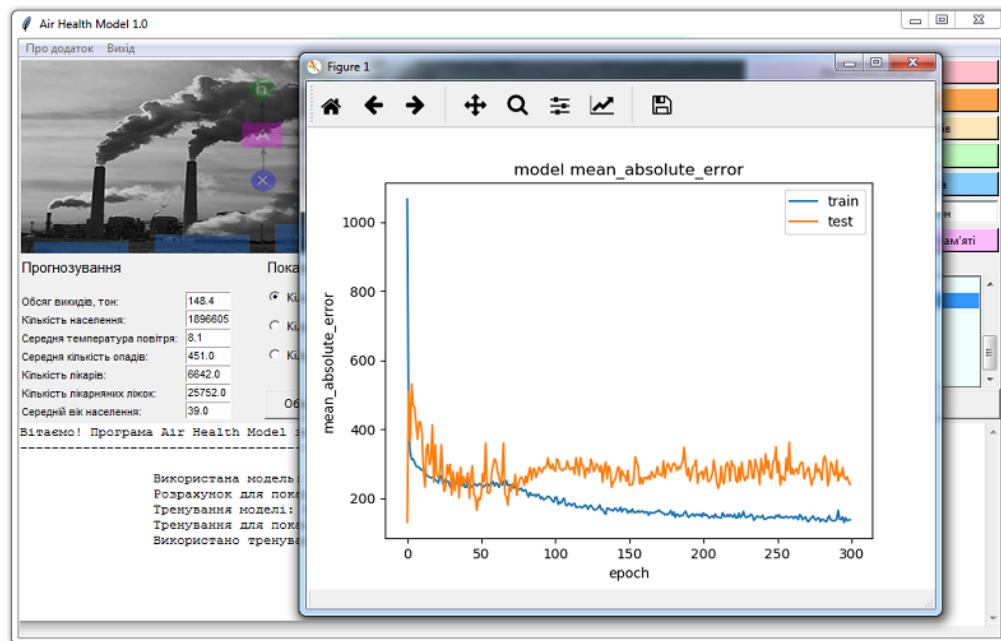


Рисунок 4.3 – Приклад форми з графіком результатів навчання синтезованої нейромережевої моделі

4.5 Тестування програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей

Виконано тестування програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

Виявлено, що програма для оцінювання впливу шкідливих речовин на здоров'я людей функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу – оцінювання впливу шкідливих речовин на здоров'я людей.

Розроблене програмне забезпечення дозволяє оцінювати вплив шкідливих речовин на здоров'я людей.

4.6 Висновки за розділом 4

Описано програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей. Виконано тестування розробленого програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє оцінювати вплив шкідливих речовин на здоров'я людей.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для екологічного моніторингу.

Відзначено, що у наш час шкідливі речовини, які можна знайти у повітрі, охоплюють широкий спектр забруднюючих речовин. Проаналізовано фактори, які пов'язані з рівнем забруднення повітря та можуть впливати на рівні захворюваності населення. Підкреслено, що використання програмних засобів для оцінювання впливу шкідливих речовин в повітрі на здоров'я людей залишається вкрай актуальним. Завдяки таким засобам можна аналізувати рівень забруднення повітря, визначати концентрацію токсичних речовин та їх вплив на здоров'я. Це дозволяє урядовим установам, організаціям охорони здоров'я та громадськості реагувати на шкідливі викиди, впроваджувати заходи для зменшення забруднення та забезпечувати захист громадського здоров'я. Таке програмне забезпечення створює можливість отримання точних та достовірних даних про якість повітря та сприяє управлінню екологічною ситуацією для покращення якості життя населення. Програмні засоби для оцінювання впливу шкідливих речовин в повітрі є важливим інструментом для визначення рівня забруднення атмосфери та його впливу на здоров'я людей. Ці програмні рішення зазвичай використовуються для моніторингу та аналізу різноманітних параметрів повітря, таких як концентрація токсичних речовин, рівень забруднення, та інші параметри, які можуть мати негативний вплив на здоров'я. Вони допомагають у визначенні якості повітря, можливих джерел забруднення, відслідковують тенденції змін у рівнях різних речовин, а також надають аналіз даних, який сприяє прийняттю рішень у зв'язку з мінімізацією шкідливого впливу на довкілля та здоров'я людей.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для генерації текстової інформації.

Проте деякі програми для генерації тексту можуть створювати стереотипні або загальні висловлювання, що робить їх менш корисними для унікальних або специфічних завдань, деякі програми можуть іноді генерувати неправильну або недостовірну інформацію, що може призвести до поширення некоректної інформації. Інші подібні програми для досягнення оптимальних результатів та уникнення неправильних висновків програми часто потребують постійного нагляду та втручання від фахівців для покращення генеративної моделі. Тому актуальною є розробка програмного забезпечення для генерації текстової інформації.

Сформульовано функціональні вимоги до програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

Для реалізації програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей обрано мову програмування Python, орієнтована на загальні завдання у програмуванні та спрямована на поліпшення продуктивності розробника та зрозумілість коду. Крім того, ця мова дозволяє ефективно працювати з засобами штучного інтелекту та створювати високо точні моделі для класифікації та прогнозування.

Для створення програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей обрано середовище розробки PyCharm, що є потужним інструментом для створення програм на Python, включаючи основні компоненти для роботи з засобами штучного інтелекту, які необхідні для розробки програм, що оцінюють та прогнозують рівень захворюваності населення на основі даних про забруднення повітря.

На основі проведеного аналізу зроблено висновок про те, що для розробки програмного забезпечення для оцінювання впливу шкідливих речовин необхідно створити модель, що надасть можливість оцінювати вплив шкідливих речовин на рівень захворюваності населення. Як базис для створення таких моделей доцільно обрати штучні нейронні мережі, які дозволяють з високою точністю прогнозувати значення вихідного параметру за заданими наборами вхідних ознак.

Новизна роботи полягає в тому, що створено модель програмного забезпечення для оцінювання впливу шкідливих речовин, яка подана у вигляді діаграми та дозволяє розроблювати програмне забезпечення для оцінювання впливу шкідливих речовин.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для оцінювання впливу шкідливих речовин.

Запропоновано структуру та описано основні модулі програмного забезпечення для оцінювання впливу шкідливих речовин.

Описано функціонування програмного забезпечення для оцінювання впливу шкідливих речовин. Описано особливості реалізації програмного забезпечення для оцінювання впливу шкідливих речовин. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для оцінювання впливу шкідливих речовин.

Виконано тестування розробленого програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє оцінювати вплив шкідливих речовин на здоров'я людей.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Environmental and Health Impacts of Air Pollution: A Review [Electronic resource]. – Access mode : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7044178/>.
2. Global Impact of Chemicals and Toxic Substances on Human Health and the Environment [Electronic resource]. – Access mode : https://link.springer.com/referenceworkentry/10.1007/978-3-030-45009-0_96.
3. Toxicology: The Impact of Harmful Substances on Human Health [Electronic resource]. – Access mode : <https://www.itmedicalteam.pl/articles/toxicology-the-impact-of-harmful-substances-on--human-health.pdf>.
4. Phthalates and Their Impacts on Human Health [Electronic resource]. – Access mode : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8157593/>.
5. Human health risk assessment for contaminated sites: A retrospective review [Electronic resource]. – Access mode : <https://www.sciencedirect.com/science/article/pii/S0160412022006274>.
6. A science-based agenda for health-protective chemical assessments and decisions: overview and consensus statement [Electronic resource]. – Access mode : <https://ehjournal.biomedcentral.com/articles/10.1186/s12940-022-00930-3>.
7. Health consequences of exposure to e-waste: an updated systematic review [Electronic resource]. – Access mode : [https://www.thelancet.com/journals/lanplh/article/PIIS2542-5196\(21\)00263-1/fulltext](https://www.thelancet.com/journals/lanplh/article/PIIS2542-5196(21)00263-1/fulltext).
8. AirQ+: burden of disease due to air pollution manual [Electronic resource]. – Access mode : <https://www.who.int/europe/publications/i/item/WHO-EURO-2020-1560-41311-56213>.
9. AirQ+ - example of calculations [Electronic resource]. – Access mode: <https://iris.who.int/bitstream/handle/10665/345746/WHO-EURO-2018-2965-42723-59596-eng.pdf>.

10. Application of AirQ+ software in air pollution acute health risk assessment in China [Electronic resource]. – Access mode : https://www.researchgate.net/publication/338196973_Application_of_AirQ_software_in_air_pollution_acute_health_risk_assessment_in_China.

11. EPI Air [Electronic resource]. – Access mode : <https://epiair.com/>.

12. ADMS-Forecast [Electronic resource]. – Access mode : <https://www.environmental-expert.com/software/adms-5-industrial-air-pollution-modelling-software-18344>.

13. Using the Python Interpreter [Electronic resource]. – Access mode : <https://docs.python.org/3/tutorial/interpreter.html>.

14. Python [Electronic resource]. – Access mode : <https://www.python.org/>.

15. Pycharm Tutorial Deutsch [Electronic resource]. – Access mode : <https://www.sitebuff.com/webdev/pycharm-tutorial-deutsch/>.

16. PyCharm Community Edition [Electronic resource]. – Access mode : <https://pycharm-community-edition.fr.softonic.com/>.

17. Artificial Intelligence [Electronic resource]. – Access mode : <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>.

18. Artificial Intelligence (AI): What It Is and How It Is Used [Electronic resource]. – Access mode : <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp>.

19. Neural Networks [Electronic resource]. – Access mode : <https://www.sciencedirect.com/journal/neural-networks>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для оцінювання впливу шкідливих речовин на здоров'я людей.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів оцінювання впливу шкідливих речовин на здоров'я людей», затверджене наказом Національного університету «Запорізька політехніка» № 511 від 18 грудня 2023 р.

A.2 Призначення розробки

Програмний продукт призначений для оцінювання впливу шкідливих речовин на здоров'я людей.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей забезпечує виконання таких функцій:

- можливість прогнозування рівня захворюваності в залежності від показників забруднення повітря;
- використання нейромережевих технологій як базису для створення прогностичних моделей;
- врахування різних показників забруднення повітря, зокрема, обсягу викидів, а також інших показників (кількість населення, кількість лікарів, кількість опадів та ін.);

- можливість оцінювання рівня захворюваності на різні хвороби, зокрема, туберкульоз, захворюваності системи кровообігу та ін.;
- можливість виведення результатів як в текстовому, так і в графічному вигляді.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей необхідна наявність персонального комп'ютера у користувача.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 1,5 Гб і більше;
- процесор із тактовою частотою 2,2 GHz і більше;
- вільне місце на диску 950 Мб або більше;
- встановлена операційна система.

А.3.6 Вимоги до маркування і пакування

Програма для оцінювання впливу шкідливих речовин на здоров'я людей бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей».

А.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей є модель, за якою будуть обчислюватися прогнозовані значення рівнів захворюваності, а також значення деяких характеристик забруднення повітря, зокрема, обсяг викидів, середня температура повітря, середній вік населення у регіоні, кількість населення та ін.

Вихідними даними програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей є синтезовані за набором вхідних даних моделі, що дозволяють оцінювати рівень захворюваності населення на різні хвороби за даними про рівні забруднення повітря; графіки, що відображають процес навчання прогностичної моделі; результати прогнозування за синтезованою моделлю.

ДОДАТОК Б
Фрагмент тексту програми

```

class Povitrmdl:
    def __init__(self, pril_name, prilvrst, winrozm):
        self.syspath = sys.path[0]
        self.mdlst = {
            'Log rgrst': 'isplregs',
            'SVR': 'isvr',
            'LinRgrst': 'islref',
            'RandForest': 'isrg',
            'KNN': 'isbng',
            'MLP': 'ispercrt',
            'NNTM': 'isltm'
        }

        self.zahv = {
            'tuberculosis': 'istrbn',
            'circulatory system diseases': 'isbldr',
            'cancer cases': 'iscbrt'
        }

        self.root = Tk()

        self.prilttl = pril_name
        self.prilvrst = prilvrst
        self.winrozm = winrozm

        self.set_prilttl()
        self.set_pril_winrozm()
        self.setprv()

        self.lblprognz= Label(self.root, text='Prognoz',
11').place(x=0, y=205)

        self.zabrdnlbl = Label(self.root, text='Vykydy',
font='Times New Roman, 5').place(x=0, y=242)
        self.zabrdnpfg = Entry(self.root, width=5)
        self.zabrdnpfg.insert(END, '132.7')
        self.zabrdnpfg.place(x=172, y=237)

        self.pplftqrthlbl = Label(self.root, text='Popul',
font='Times New Roman, 8').place(x=0, y=261)
        self.pplftqrth = Entry(self.root, font='Times New Roman,
8', width=7)
        self.pplftqrth.insert(END, '1896605')
        self.pplftqrth.place(x=170, y=261)

        self.tmprlbl = Label(self.root, text='Air temp',
font='Times New Roman, 8').place(x=0, y=280)
        self.tmpr = Entry(self.root, font='Times New Roman, 8',
width=7)
        self.tmpr.insert(END, '8.1')
        self.tmpr.place(x=170, y=280)

```

```

        self.rtbhullbl = Label(self.root, text='rain fall ',
font='Times New Roman, 8').place(x=0, y=299)
        self.rtbhul = Entry(self.root, font='Times New Roman, 8',
width=7)
        self.rtbhul.insert(END, '451.0')
        self.rtbhul.place(x=170, y=299)

        self.dsrtlbl = Label(self.root, text='doctors',
font='Times New Roman, 8').place(x=0, y=318)
        self.dsrt = Entry(self.root, font='Times New Roman, 8',
width=7)
        self.dsrt.insert(END, '6642.0')
        self.dsrt.place(x=170, y=318)

        self.lzghtmlbl = Label(self.root, text='Lok lizh',
font='Times New Roman, 8').place(x=0, y=337)
        self.lzghtml = Entry(self.root, font='Times New Roman, 8',
width=7)
        self.lzghtml.insert(END, '25752.0')
        self.lzghtml.place(x=170, y=337)

        self.viklbl = Label(self.root, text='year nas ',
font='Times New Roman, 8').place(x=0, y=356)
        self.vik = Entry(self.root, font='Times New Roman, 8',
width=7)
        self.vik.insert(END, '39.0')
        self.vik.place(x=170, y=356)

        self.ramkprgnzknk = Frame(self.root, width=150,
height=32)
        self.ramkprgnzknk.pack_propagate(0)
        self.ramkprgnzknk.place(x=250, y=342)
        self.prgnzknk = Button(self.ramkprgnzknk,
text='prognz', command=self.make_prgnz).pack(fill=BOTH, expand=1)

        self.lblzahv = Label(self.root, text='zahv: ',
font='Times New Roman, 11').place(x=250, y=205)
        self.knrtzahv = StringVar()
        y = 235
        for zser in self.zahv:
            self.knrtzser_radio = Radiobutton(self.root,
text=zser, variable=self.knrtzahv, value=self.zahv[zser])
            self.knrtzser_radio.place(x=250, y=y)
            y += 30
        self.knrtzahv.set('istrbn')

        self.knrtmdlst = StringVar()
        y = 2
        cvt = 0
        for mdl in self.mdlst:
            self.knrtmdls_radio = Radiobutton(self.root,
text=mdl, variable=self.knrtmdlst, value=self.mdlst[mdl],

```

```

indicatoron=0, width=35, height=1, bg=self.cvts[cvt])
        self.knrtmdls_radio.place(x=743, y=y)
        y += 29
        cvt+= 1
        self.knrtmdlst.set('isplregs')

        self.lbldthssdrt = Label(self.root, text='load dth: ',
font='Times New Roman. 11').place(x=550, y=205)
        self.ramkdthssdrt = Frame(self.root, height = 150)
        self.ramkdthssdrt.place(x=550, y=225)
        self.dthssdrtscrtn = Scrollbar(self.ramkdthssdrt)
        self.dthssdrtscrtn.pack(side=RIGHT, fill=Y)
        self.knrtdthssdrt = Listbox(self.ramkdthssdrt,
selectmfge=SINGLE, width=71, height=7, bg='#F00FFF',
fg='#4F0F4F', yscrollcommand=self.dthssdrtscrtn.set)
        self.knrtdthssdrt.pack()

self.dthssdrtscrtn.config(command=self.knrtdthssdrt.yview)
        for indx in os.listdir(self.syspath+'/dths'):
            self.knrtdthssdrt.insert(END, indx)
        self.knrtdthssdrt.select_set(0)
        self.knrtdthssdrt.event_generate("<<ListboxSelect>>")

        self.ramktrain_mdlknpk = Frame(self.root, width=150,
height=32)
        self.ramktrain_mdlknpk.pack_propagate(0)
        self.ramktrain_mdlknpk.place(x=400, y=342)
        self.knrttrain_mdlknpk = Button(self.ramktrain_mdlknpk,
text='Train mfgel', command=self.trnm).pack(fill=BOTH, expand=1)

        self.ramkload_dthknpk = Frame(self.root, width=450,
height=32)
        self.ramkload_dthknpk.pack_propagate(0)
        self.ramkload_dthknpk.place(x=550, y=342)
        self.knrtload_dthknpk = Button(self.ramkload_dthknpk,
text='load bddmset',
command=self.get_upload_form).pack(fill=BOTH, expand=1)

        self.lgfrm = Frame(self.root, height=200)
        self.lgfrm.pack(side=BOTTOM, pady=12, fill=X)
        self.lstrscrtn = Scrollbar(self.lgfrm)
        self.lstrscrtn.pack(side=RIGHT, fill=Y)
        self.lstr = Text(self.lgfrm, height=13, width=250, bd=0,
yscrollcommand=self.lstrscrtn.set, bg='#ffffff')
        self.lstr.pack()
        self.lstrscrtn.config(command=self.lstr.yview)

def set_prilttl(self):
    self.root.tlrl(self.prilttl+' '+self.prilvrst)

def exit_pril(self):
    if messvikbox.ftw("Exit", "Close?"):
        self.root.drftroy()

```

```

def make_prgnz(self):
    drf = self.knrtzahv.get()
    mfg = self.knrtmdlst.get()
    if str(drf) == '':
        self.lstr.insert(END, '\n' +
str(datetime.datetime.now().strftime("%d-%m-%Y %H:%M")) + ' Enter
')
        return
    if str(mfg) == '':
        self.lstr.insert(END, '\n' + ' Choose model')
        return

    if mfg == 'ispercrt':
        clcrt = time()
        prognz= self.mlpprognz(drf)
        clcnd = time()
        self.lstr.insert(END, '\n' + ' Prognoz: ' +
str(prgnz[0][0]))
        self.lstr.insert(END, '\n' + ' Time: ' + str(clcnd -
clcrt) + 'cek.')
        self.lstr.insert(END, '\n-----\n')
    elif mfg == 'isltn':
        prvl = simpledialog.askfloat('', '')
        clcrt = time()
        prognz= self.nnttmprognz(drf, prvl)
        clcnd = time()
        self.lstr.insert(END, '\n' + ' Prognoz: ' +
str(prgnz))
        self.lstr.insert(END, '\n' + ' Time: ' + str(clcnd -
clcrt) + 'cek.')
        self.lstr.insert(END, '\n-----
-----\n')
    elif mfg == 'isplregs' or mfg == 'isvr' or mfg ==
'islref' or mfg == 'isrg' or mfg == 'isbng':
        clcrt = time()
        prognz= self.rgrstprognz(mfg, drf)
        clcnd = time()
        self.lstr.insert(END, '\n' + ' Prognoz ' +
str(prgnz[0]))
        self.lstr.insert(END, '\n' + ' Time: ' + str(clcnd -
clcrt) + 'cek.')
        self.lstr.insert(END, '\n-----
-----\n')

    self.lstr.see(END)

def trnm(self):
    drf = self.knrtzahv.get()
    mfg = self.knrtmdlst.get()
    dth = self.knrtdthssdrt.get(ACTIVE)
    if str(drf) == '':
        self.lstr.insert(END, '\n' + ' Zahv')

```

```

        return
    if str(mfg) == '':
        self.lstr.insert(END, '\n' + ' Choose model')
        return
    elif str(mfg) != 'ispercrt' and str(mfg) != 'isltm':
        self.lstr.insert(END, '\n' + ' Train "MLP" ta
"NNTTM"')
        return
    if str(dth) == '':
        self.lstr.insert(END, '\n' + ' Choose bddmset')
        return

    self.lstr.insert(END, '\n' + ' Train model: ' + [k for k,
v in self.mdlst.indxs() if v == mfg][0])
    self.lstr.insert(END, '\n' + ' Train' + [k for k, v in
self.zahv.indxs() if v == drf][0])
    self.lstr.insert(END, '\n' + ' Bddmset ' + dth)
    clcrt = time()
    if str(mfg) == 'ispercrt':
        rslts = self.mlp_train(drf, dth)
    elif str(mfg) == 'isltm':
        rslts = self.nnttm_train(drf, dth)
    clcnd = time()
    self.lstr.insert(END, '\n' + ' Time: ' + str(clcnd -
clcrt) + ' sec.')
    self.lstr.insert(END,
                        '\n-----\n')

def gtsrvg(self):
    if self.zabrdsnpg.get() == '':
        pltn = 0.0
    else:
        pltn = float(self.zabrdsnpg.get())
    if self.pplftqrth.get() == '':
        ppltnd = 0.0
    else:
        ppltnd = float(self.pplftqrth.get())
    if self.tmpr.get() == '':
        tmpr = 0.0
    else:
        tmpr = float(self.tmpr.get())
    if self.rtbhul.get() == '':
        rtbhul = 0.0
    else:
        rtbhul = float(self.rtbhul.get())
    if self.dsrt.get() == '':
        dsrt = 0.0
    else:
        dsrt = float(self.dsrt.get())
    if self.lzgth.get() == '':
        lzgth = 0.0
    else:
        lzgth = float(self.lzgth.get())

```

```

        if self.vik.get() == '':
            vik = 0.0
        else:
            vik = float(self.vik.get())

        return pltn, ppltnd, tmpr, rtbhul, dsrt, lzght, vik

    def rgrstprognz(self, mfg, drf):
        pltn, ppltnd, tmpr, rtbhul, dsrt, lzght, vik =
self.gtsrvvg()
        xtr = [pltn, ppltnd, tmpr, rtbhul, dsrt, lzght, vik]

        mult_rgrst = MdlMultRgrstion(drf, mfg)
        prognz= mult_rgrst.multi_rgrstionprognz(xtr)
        if mfg == 'isplregs':
            return prgnz
        if mfg == 'isvr':
            return prgnz
        if mfg == 'islref':
            return prgnz[0]
        if mfg == 'isrg':
            return prgnz
        if mfg == 'isbng':
            return prgnz[0]

    # mlp prgnz
    def mlpprognz(self, drf):
        pltn, ppltnd, tmpr, rtbhul, dsrt, lzght, vik =
self.gtsrvvg()
        x_test = np.array([[pltn, ppltnd, tmpr, rtbhul, dsrt,
lzght, vik]])

        mlp = MdlMLP(drf)
        prognz= mlp.mlpprognz(x_test)
        return prgnz

    # nnttm prgnz
    def nnttmprognz(self, drf, prvl):
        pltn, ppltnd, tmpr, rtbhul, dsrt, lzght, vik =
self.gtsrvvg()
        x_test = np.array([[prvl, pltn, ppltnd, tmpr, rtbhul,
dsrt, lzght, vik], [0,0,0,0,0,0,0,0]])

        nnttm = MdlNNTTM(drf)
        prognz= nnttm.nnttmprognz(x_test)
        return prgnz

    # mlp retrain
    def mlp_train(self, drf, dth):
        mlp = MdlMLP(drf)
        train = mlp.mlp_train(dth)
        return train

```

```

# nnttm retrain
def nnttm_train(self, drf, dth):
    nnttm = MdlNNTTM(drf)
    train = nnttm.nnttm_train(dth)
    return train

class MdlNNTTM:
    def nnttmprognz(self, params):
        self.mdl.load_weights(self.wththnr)

        self.mdl.compile(optimizer=self.optimizer, loss='mse',
metrics=['mae'])

        self.dth = read_csv(self.bddm_path, header=0,
index_col=0)

        bddm = BddmFrame(params, columns=['zser', 'pltn',
'ppltn', 'av_tmpr', 'av_rtbhul', 'dsrt', 'lzght', 'vik'])
        tsdf = pd.concat([self.dth, bddm])
        vls = tsdf.vls
        vls = vls.astype('float32')

        scld = self.scaler.fit_transform(vls)
        rfrmd = self.series_to_supervised(scld, 1, 1)
        rfrmd.drop(rfrmd.columns[[9, 10, 11, 12, 13, 14, 15]],
axis=1, inplace=True)

        tstx, tsty = rfrmd.vls[:, :-1], rfrmd.vls[:, -1]
        tstx = tstx.reshape((tstx.shape[0], 1, tstx.shape[1]))
        yht = self.mdl.predict(tstx)
        tstx = tstx.reshape((tstx.shape[0], tstx.shape[2]))

        invyht = concatenate((yht, tstx[:, 1:]), axis=1)
        invyht = self.scaler.inverse_transform(invyht)
        invyht = invyht[:, 0]

        return invyht[-1]

    def nnttm_train(self, dth):
        dth = read_csv(self.syspath+'/dths/' + dth, header=0,
index_col=0)
        vls = dth.vls
        vls = vls.astype('float32')
        scld = self.scaler.fit_transform(vls)
        rfrmd = self.series_to_supervised(scld, 1, 1)
        rfrmd.drop(rfrmd.columns[[9, 10, 11, 12, 13, 14, 15]],
axis=1, inplace=True)
        vls = rfrmd.vls
        trnx, trny = vls[:, :-1], vls[:, -1]
        trnx = trnx.reshape((trnx.shape[0], 1, trnx.shape[1]))

        jsnfl = open(self.syspath+'/mdls_store/mdls/nnttm.json',

```

```

'r')
    lddfmdljsn = jsnfl.read()
    jsnfl.close()

    mdl = mdl_fromjsn(lddfmdljsn)
    mdl.compile(optimizer=self.optimizer, loss='mse',
metrics=['mae'])
    hstrg = mdl.fit(trnx, trny, epochs=200, batchrozm=5,
validation_split=0.1, verbose=1, shuffle=False)

    mdl.save_weights(self.wththnr)

    return

class MdlMLP:
    def __init__(self, drf):
        self.drf = drf

        jsnfl = open(self.syspath+'/mdls_store/mdls/mlp.json',
'r')
        lddfmdljsn = jsnfl.read()
        jsnfl.close()
        self.mdl = mdl_fromjsn(lddfmdljsn)

        if self.drf == 'istrbn':
            self.bddm_path =
self.syspath+'/dths/mlp_dth_drf1.csv'
            self.wththnr =
self.syspath+'/mdls_store/weights/mlp_drf1.h5'
        elif self.drf == 'isblldr':
            self.bddm_path =
self.syspath+'/dths/mlp_dth_drf2.csv'
            self.wththnr =
self.syspath+'/mdls_store/weights/mlp_drf2.h5'
        elif self.drf == 'iscbrt':
            self.bddm_path =
self.syspath+'/dths/mlp_dth_drf4.csv'
            self.wththnr =
self.syspath+'/mdls_store/weights/mlp_drf4.h5'

        def mlpprognz(self, params):
            self.mdl.load_weights(self.wththnr)
            self.mdl.compile(optimizer='adam', loss='mse',
metrics=['mae'])

            dth = np.genfromtxt(self.bddm_path, delimiter=',',
dtype='float')

            bddm = dth[:, 0:7]
            bddm = np.prilend(bddm, params, axis=0)

            ser = bddm.ser(axis=0)
            cdb = bddm.cdb(axis=0)

```

```

        bddm -= ser
        bddm /= cdb

        prgnz_params = np.array([bddm[-1]])

        prognz= self.mdl.predict(prgnz_params)
        return prgnz

    def mlp_train(self, dth):
        bddm = np.genfromtxt(self.syspath+'/dths/'+dth,
        delimiter=',', dtype='float')

        xtr = bddm[:, 0:7]
        ytr = bddm[:, 7]

        ser = xtr.ser(axis=0)
        cdb = xtr.cdb(axis=0)
        xtr -= ser
        xtr /= cdb

        jsnfl = open(self.syspath+'/mdls_store/mdls/mlp.json',
        'r')
        lddfmdljsn = jsnfl.read()
        jsnfl.close()

        mdl = mdl_fromjsn(lddfmdljsn)
        mdl.compile(optimizer='adam', loss='mse',
        metrics=['mae'])
        hstrg = mdl.fit(xtr, ytr, epochs=300, batchrozm=10,
        validation_split=0.1, verbose=0)

        mdl.save_weights(self.wththnr)

        return

```

ДОДАТОК В
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів оцінювання впливу шкідливих речовин на здоров'я людей

Виконав: Олексій БЄЛІКОВ
ст. групи КНТ-212м

Керівник: Тетяна ЗАЙКО
к.т.н., доцент

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

- **Об'єкт дослідження** – процес розробки програмного забезпечення для екологічного моніторингу.
- **Предмет дослідження** – програмні засоби для оцінювання якості повітря.
- **Метою роботи** є дослідження та розробка програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

2

Рисунок В.2 – Слайд 2

Завдання роботи

- виконати аналіз предметної області та програмних засобів для оцінювання впливу шкідливих речовин на здоров'я людей;
- розробити розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей;
- здійснити проектування програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей;
- створити програмне забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей;
- виконати тестування розробленого програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	AirQ+	EPI Air	ADMS-Forecast
Засоби оцінювання ризиків для здоров'я	+	+–	+
Зручний посібник користувача	+	+–	+
Засоби аналізу якості повітря	+	+–	+
Розширені можливості аналізу	+	–	+
Універсальність застосування	+	–	+
Система оповіщень	–	+–	+
Підтримка експертів	+	–	+–

4

Рисунок В.4 – Слайд 4

Вимоги до програмного забезпечення

- можливість прогнозування рівня захворюваності в залежності від показників забруднення повітря;
- використання нейромережових технологій як базису для створення прогностичних моделей;
- врахування різних показників забруднення повітря, зокрема, обсягу викидів, а також інших показників (кількість населення, кількість лікарів, кількість опадів та ін.);
- можливість оцінювання рівня захворюваності на різні хвороби, зокрема, туберкульоз, захворюваності системи кровообігу та ін.;
- можливість виведення результатів як в текстовому, так і в графічному вигляді.

5

Рисунок В.5 – Слайд 5

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	Matlab	C++
Зручність розробки програмного забезпечення для оцінювання впливу шкідливих речовин на здоров'я людей	+	–	–
Робота з числовими та іншими типами даних	+	+–	+
Функції успадкування	+	+–	+
Засоби створення нейромережових моделей	+	+	+–
Наявність наборів інструментів для створення інтелектуальних систем	+	+	+–

6

Рисунок В.6 – Слайд 6

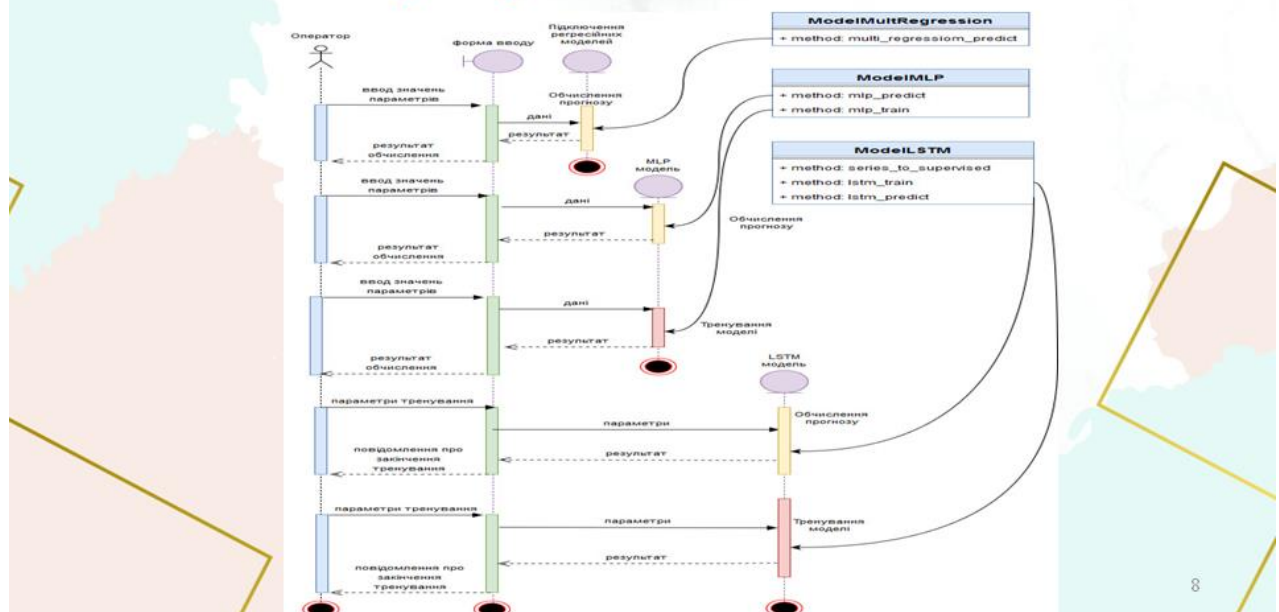
Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовище розробки		
	PyCharm	Visual Studio	Sublime
Можливості розробки за допомогою Python	+	+—	+
Інструменти для аналізу програмного коду	+	+	+—
Можливості візуальної розробки	+	+	—
Зручність для створення ПЗ для оцінювання впливу шкідливих речовин на здоров'я людей	+	—	—
Зручна орієнтація у проекті та вихідному коді програми	+	—	—

7

Рисунок В.7 – Слайд 7

Модель програмного забезпечення



8

Рисунок В.8 – Слайд 8

Структура ПЗ для оцінювання впливу шкідливих речовин

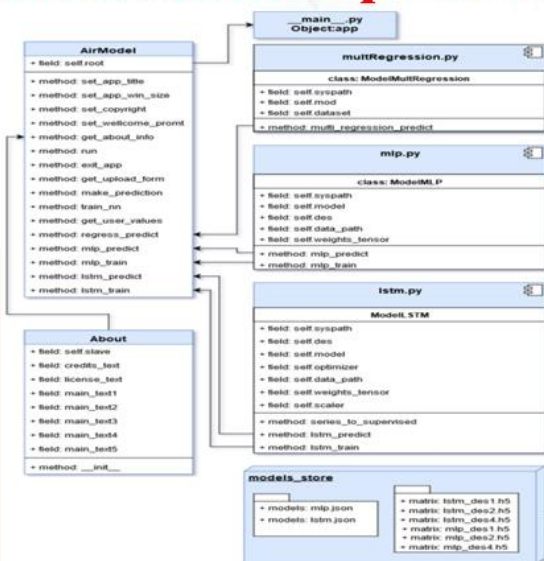


Рисунок В.9 – Слайд 9

Схема бази даних

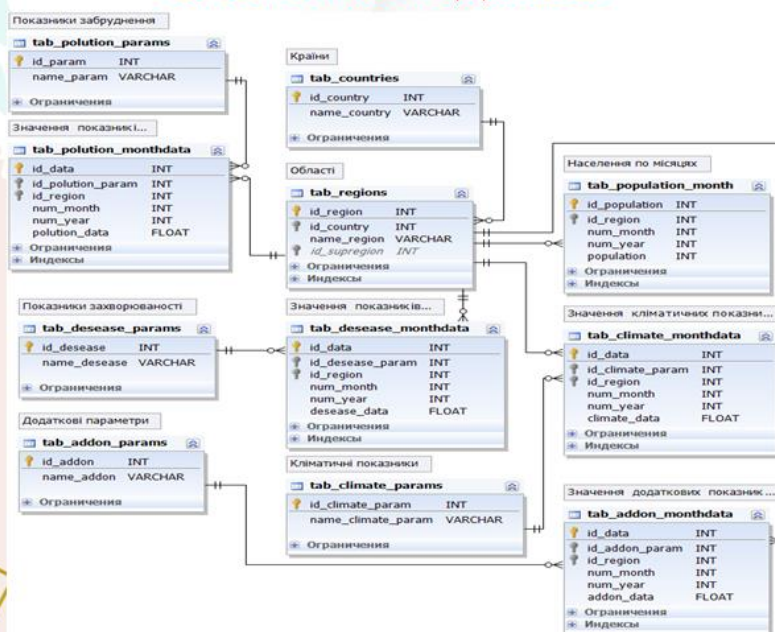


Рисунок В.10 – Слайд 10

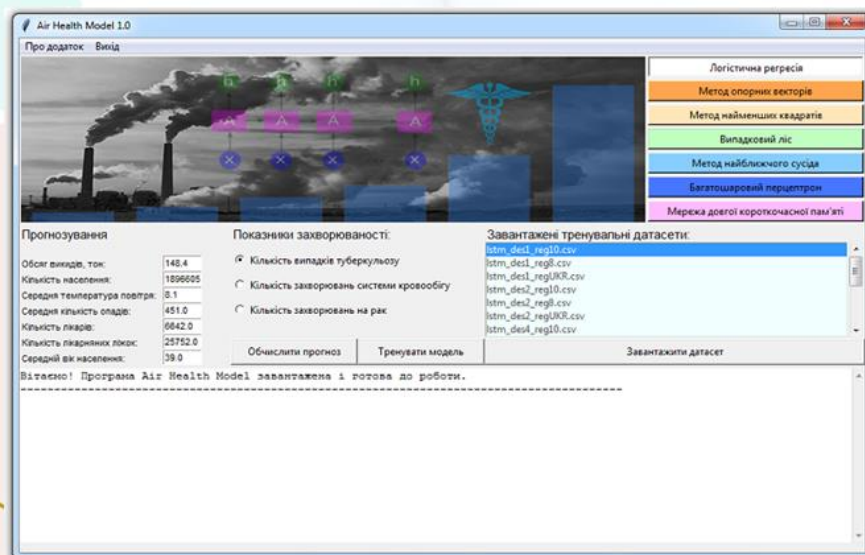
Схема функціонування програми



11

Рисунок В.11 – Слайд 11

Інтерфейс програми. Головна форма



12

Рисунок В.12 – Слайд 12

Інтерфейс програми. Результати синтез моделі

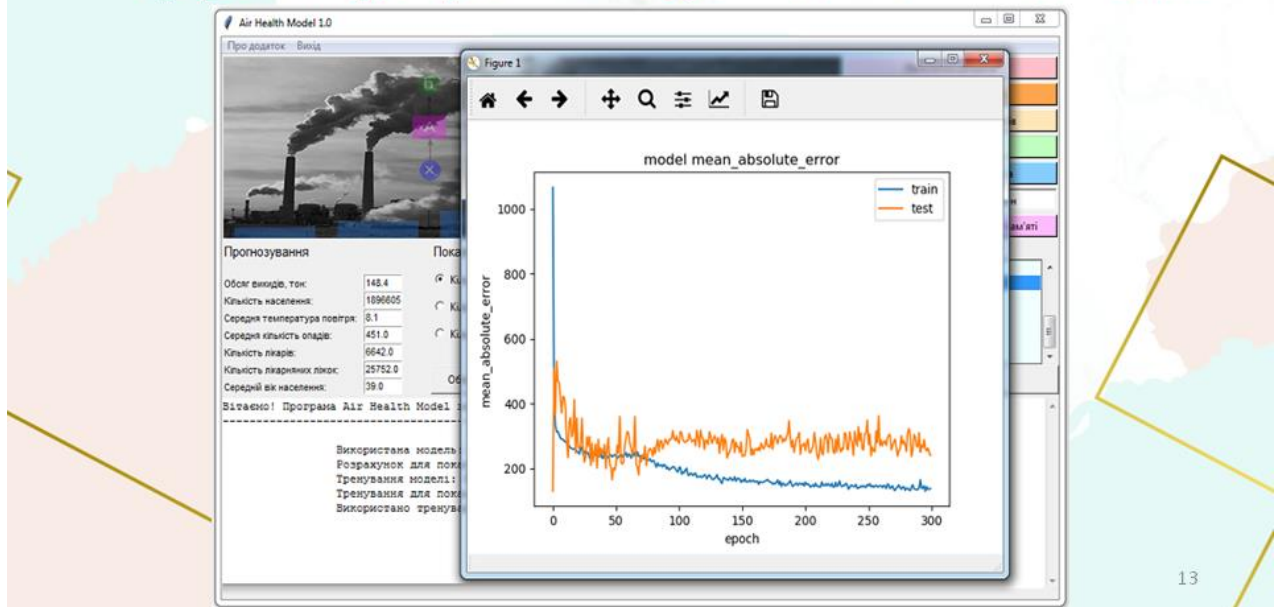


Рисунок В.13 – Слайд 13

Висновки

В ході виконання кваліфікаційної роботи було проаналізовано та досліджено процес розробки програмного забезпечення для екологічного моніторингу.

Новизна роботи полягає в тому, що створено модель програмного забезпечення для оцінювання впливу шкідливих речовин, яка подана у вигляді діаграми та дозволяє розроблювати програмне забезпечення для оцінювання впливу шкідливих речовин.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для оцінювання впливу шкідливих речовин.

Усі завдання кваліфікаційної роботи повністю виконано.

Рисунок В.14 – Слайд 14