

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра інформаційно-обчислювальних систем і управління

КАСЬЯН Тамара Володимирівна

**Програмний модуль аналізу і прогнозування інфекційних
захворювань з використанням методів машинного навчання /
Software module for analysis and forecasting of infectious diseases
using machine learning methods**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студентка групи КН-41
Т. В. Касьян

Науковий керівник:
к.т.н., доцент І. В. Турченко

Кваліфікаційну роботу допущено до
захисту

«___»_____ 2025 р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
« ____ » _____ 2024р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
КАСЬЯН Тамара Володимирівна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Програмний модуль аналізу і прогнозування інфекційних захворювань з використанням методів машинного навчання / Software module for analysis and forecasting of infectious diseases using machine learning methods

керівник роботи к.т.н., доцент І.В. Турченко

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

затверджені наказом по університету від 20 грудня 2024 р. № 938.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- провести аналіз предметної області;
- проаналізувати існуючі аналоги та зробити постановку задачі проектування;
- представити загальну структуру та алгоритмічне забезпечення програмного модуля;
- розробити програмний модуль прогнозування, що реалізує обрані моделі машинного навчання та забезпечує формування результатів у вигляді графіків;
- розробити інтерфейс користувача;
- протестувати роботу модуля і провести експериментальні дослідження точності прогнозування.

5. Перелік графічного матеріалу в роботі:

- дерево функцій модуля прогнозування інфекційних захворювань;
- IDEF0 для модуля прогнозування;
- діаграма потоку даних для модуля прогнозування;
- схема потоків даних в модулі.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студент _____ Т.В. Касьян
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ І.В. Турченко
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль аналізу і прогнозування інфекційних захворювань з використанням методів машинного навчання» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом у 66 сторінок і містить 13 ілюстрацій, 1 таблицю, 3 додатки та 25 використаних джерел.

Метою кваліфікаційної роботи є створення програмного модуля, що дозволяє аналізувати та прогнозувати динаміку інфекційних захворювань на основі історичних даних із застосуванням методів машинного навчання, з інтерактивною візуалізацією результатів для підтримки прийняття управлінських рішень у сфері охорони здоров'я.

Об'єктом дослідження є процес прогнозування поширення інфекційних захворювань.

Предметом дослідження є застосування моделей машинного навчання (ARIMA, SARIMA, Prophet, Stacking та ін.) для побудови прогнозів на основі медичних статистичних даних, а також програмна реалізація аналітичного модуля з інтерактивним інтерфейсом користувача.

У межах роботи реалізовано повноцінну систему збору, зберігання, обробки та візуалізації даних. Створено базу даних у MySQL, розроблено модулі обробки даних і прогнозування у Python, а також веб-інтерфейс із застосуванням HTML, CSS, JavaScript і бібліотеки Plotly. Проведено експериментальне порівняння точності прогнозів різних моделей, тестування модуля та розміщення його у веб-середовищі.

Ключові слова: ПРОГНОЗУВАННЯ, ІНФЕКЦІЙНІ ЗАХВОРЮВАННЯ, МАШИННЕ НАВЧАННЯ, PYTHON, MYSQL, ВІЗУАЛІЗАЦІЯ, СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ.

ANNOTATION

The bachelor's qualification thesis entitled “Software Module for the Analysis and Forecasting of Infectious Diseases Using Machine Learning Methods”, completed to obtain the degree of Bachelor in the specialty 122 “Computer Science” under the educational program “Computer Science,” consists of 66 pages and includes 13 figures, 1 table, 3 appendices, and 25 references.

The aim of this qualification thesis is to develop a software module for analyzing and forecasting the dynamics of infectious diseases based on historical data using machine learning methods, with interactive visualization of results to support decision-making in the field of public health.

The object of the research is the process of forecasting the spread of infectious diseases.

The subject of the research is the application of machine learning models (ARIMA, SARIMA, Prophet, Stacking, etc.) for building forecasts based on medical statistical data, as well as the software implementation of an analytical module with an interactive user interface.

Within the scope of the thesis, a complete system for data collection, storage, processing, and visualization was implemented. A MySQL database was created, data processing and forecasting modules were developed using Python, and a web interface was built using HTML, CSS, JavaScript, and the Plotly library. An experimental comparison of forecast accuracy across different models was conducted, alongside comprehensive module testing and deployment in a web environment.

Keywords: FORECASTING, INFECTIOUS DISEASES, MACHINE LEARNING, PYTHON, MYSQL, VISUALIZATION, DECISION SUPPORT SYSTEM.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області	9
1.1 Опис предметної області	9
1.2 Огляд та аналіз існуючих рішень	13
1.3 Постановка задачі.....	17
2 Алгоритмічне та інформаційне забезпечення модуля аналізу та прогнозування інфекційних захворювань	19
2.1 Загальна структура проектного модуля аналізу та прогнозування інфекційних захворювань	19
2.2 Алгоритмічне забезпечення модуля аналізу та прогнозування інфекційних захворювань	22
2.3 Метрики оцінки точності прогнозувальних моделей.....	24
2.4 Інформаційне забезпечення модуля аналізу та прогнозування інфекційних захворювань	26
3 Реалізація програмного модуля аналізу та прогнозування інфекційних захворювань.....	31
3.1 Реалізація програмного забезпечення модуля аналізу та прогнозування інфекційних захворювань	31
3.2 Інтерфейс користувача модуля аналізу та прогнозування інфекційних захворювань	36
3.3 Тестування розробленого модуля	38
Висновки.....	43
Список використаних джерел.....	45
Додаток А Програмний код прооцесу прогнозування.....	48
Додаток Б Код користувацького інтерфейсу.....	52
Додаток В Апробація отриманих результатів	61

ВСТУП

У сучасному світі проблема прогнозування інфекційних захворювань набуває дедалі більшої актуальності. Зростаюча кількість випадків ГРВІ, ВІЛ, СНІДу та інших інфекційних хвороб вимагає від систем охорони здоров'я впровадження ефективних методів аналізу та управління епідеміологічною ситуацією. Важливим інструментом у цьому контексті стають технології машинного навчання, які дозволяють здійснювати точне прогнозування на основі історичних даних, що дає змогу завчасно вживати заходів для стримування поширення захворювань і раціонально розподіляти ресурси.

Метою роботи є створення програмного модуля аналізу і прогнозування інфекційних захворювань з використанням методів машинного навчання. Для її досягнення необхідно:

1. Провести аналіз предметної області.
2. Проаналізувати існуючі аналоги та зробити постановку задачі проєктування.
3. Представити загальну структуру та алгоритмічне забезпечення програмного модуля.
4. Розробити програмний модуль прогнозування, що реалізує обрані моделі машинного навчання та забезпечує формування результатів у вигляді графіків.
5. Розробити інтерфейс користувача.
6. Протестувати роботу модуля і провести експериментальні дослідження точності прогнозування.

Об'єктом дослідження є процес прогнозування поширення інфекційних захворювань на основі історичних статистичних даних.

Предметом дослідження є застосування методів машинного навчання для побудови моделей прогнозування та детекції.

У ході дослідження використовувалися методи класичної статистики (ARIMA, SARIMA), сучасні ансамблеві підходи (Prophet, Stacking), а також інструменти розробки програмного забезпечення - мова програмування Python. Для зберігання та обробки даних було реалізовано базу даних у середовищі MySQL.

Інтерфейс користувача, створений у Figma, забезпечує наочне представлення результатів прогнозу у вигляді інтерактивних графіків.

Результати дослідження опубліковано в матеріалах науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (ІІТАР - 2025), м. Тернопіль, 27-29 травня 2025 р. (додаток В).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Однією із ключових задач епідеміології безспірно є прогнозування поширення інфекційних захворювань. Зміни кліматичних умов, зростання глобальної мобільності, а також соціально-економічні трансформації сприяють створенню нових викликів для сучасної системи охорони здоров'я. Своєчасний аналіз статистичних даних про захворюваність дозволяє виявити та попередити потенційні епідемічні загрози, а також впровадити заходи для їх запобігання. У цій роботі розглядається процес прогнозування інфекційних захворювань із застосуванням методів машинного навчання, що сприяє як оперативному збору даних, так і їх глибокому аналітичному опрацюванню.

Для підвищення ефективності процесів прогнозування доцільно чітко розмежовувати дві основні їх складові:

а) транзакційна складова:

- 1) забезпечує автоматизований збір, накопичення та первинну обробку даних;
- 2) реалізує функції імпорту, очищення та зберігання інформації;
- 3) гарантує актуальність даних, що є базою для подальшого аналізу;

б) аналітична складова:

- 1) відповідає за побудову прогнозної моделі на основі оброблених даних;
- 2) реалізує багаторозрізний аналіз даних, що дозволяє виявляти тренди за періодами часу, типами захворювань та регіонами;
- 3) забезпечує візуальне представлення результатів для підтримки прийняття стратегічних рішень.

Для чіткого уявлення про ці складові доцільно розробити дерево функцій, яке відображає основні етапи й підетапи роботи з даними та прогнозною моделлю (рисунок 1.1).

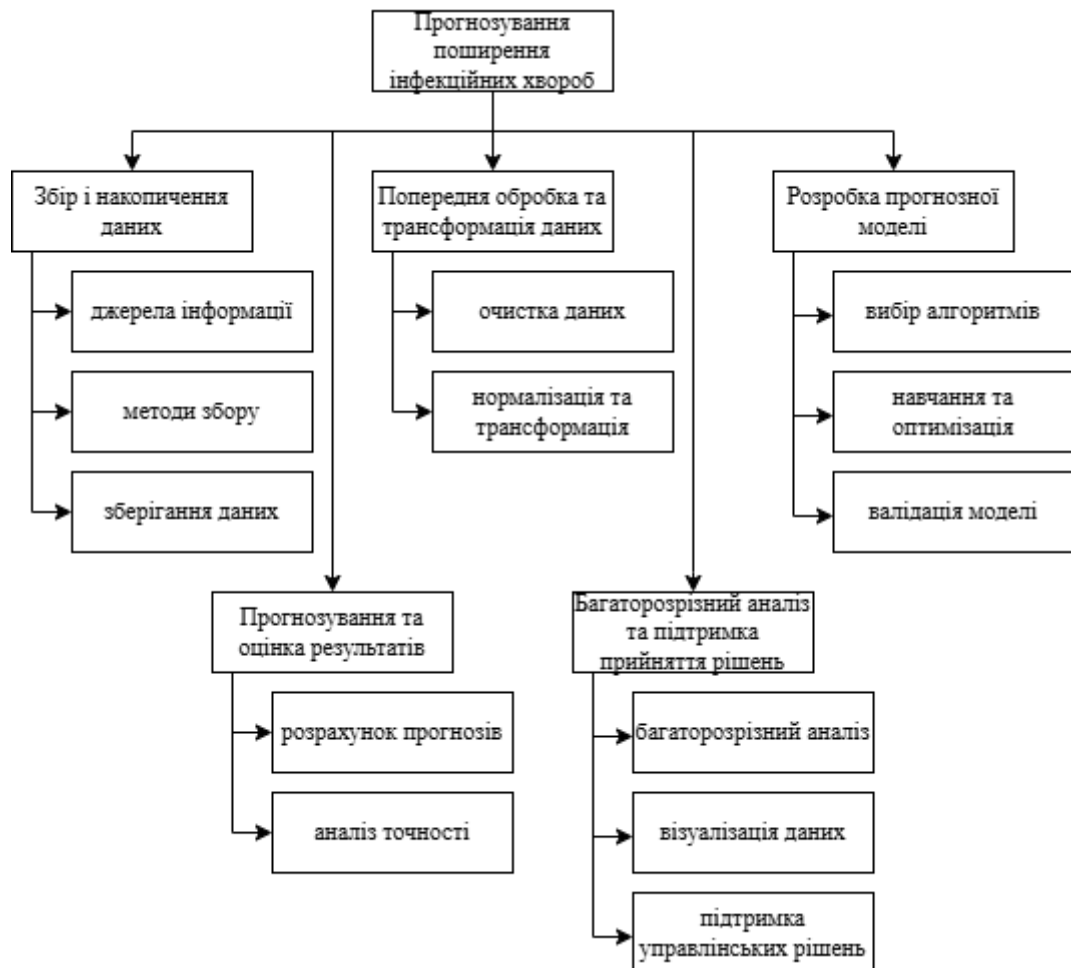


Рисунок 1.1 - Дерево функцій модуля прогнозування інфекційних захворювань

У загальному вигляді дерево функцій (див. рисунок 1.1) можна поділити на такі блоки:

а) збір та накопичення даних:

- 1) джерела інформації (дані надходять із державних відкритих інформаційних порталів охорони здоров'я);
- 2) методи збору (автоматизоване отримання даних);
- 3) зберігання даних (використання баз даних із забезпеченням архівування та резервного копіювання, що гарантує збереження історичних даних для подальшого аналізу);

б) попередня обробка та трансформація даних:

- 1) очистка даних: фільтрація некоректних, дубльованих чи неповних записів;

2) нормалізація та трансформація: перетворення зібраних даних у формат, придатний для аналізу та побудови моделей машинного навчання;

3) розробка прогнозної моделі;

в) вибір алгоритмів: оцінка різних підходів з урахуванням специфіки епідеміологічних даних;

1) навчання та оптимізація: тренування моделі на історичних даних, налаштування гіперпараметрів та використання методів крос-валідації для уникнення перенавчання;

2) валідація моделі: порівняння прогнозних показників із фактичними даними, розрахунок похибок та вдосконалення алгоритму;

г) прогнозування та оцінка результатів:

1) розрахунок прогнозів: застосування натренованої моделі для отримання прогнозних даних про майбутні епідемічні спалахи;

2) аналіз точності: регулярний аналіз відхилень прогнозу від реальних показників з метою коригування моделі та підвищення її достовірності;

д) багаторозрізний аналіз та підтримка прийняття рішень:

1) багаторозрізний аналіз: аналіз даних за часовими періодами, типами захворювань, географічними регіонами або окремими медичними установами, що дозволяє виявити сезонні коливання, локальні спалахи та інші закономірності;

2) візуалізація даних: створення інтерактивних інформаційних панелей, графіків та звітів для зручної презентації результатів аналізу експертам;

3) підтримка управлінських рішень: забезпечення інформаційної бази для оперативного реагування на загрози та планування профілактичних заходів.

Таким чином, прогнозування інфекційних захворювань охоплює сукупність процесів від збору й накопичення даних до побудови аналітичних моделей та формування підсумкових звітів. Поділ процесу прогнозування на транзакційну та аналітичну складові дає змогу ефективніше керувати інформацією та зосередитися на глибокому аналізі показників, виявленні закономірностей і своєчасному прогнозуванні потенційних епідемій.

Результати такого прогнозування дозволяють своєчасно впроваджувати профілактичні та протиепідемічні заходи.

Необхідно наголосити, що інформаційні технології відіграють дедалі важливішу роль у сфері охорони здоров'я, особливо в контексті епідеміологічного моніторингу та прогнозування інфекційних захворювань. Зі збільшенням обсягів медичних та демографічних даних, а також зростанням складності взаємозв'язків між епідеміологічними чинниками, традиційні статистичні методи дедалі частіше доповнюються або замінюються сучасними інструментами аналізу, зокрема методами машинного навчання (англ. Machine Learning (ML)).

Інструменти, які здатні обробляти великі обсяги епідеміологічних даних (наприклад, кількість захворювань за період, географічне розташування випадків, метеорологічні чинники, мобільність населення тощо), знаходити в них приховані закономірності та формувати прогнози щодо подальшого розвитку ситуації, є дуже актуальними.

Методи машинного навчання дають змогу ефективно вирішувати подібні задачі завдяки здатності моделей виявляти складні нелінійні залежності у даних, адаптуватися до нових сценаріїв та оновлювати знання у міру надходження нової інформації. Наприклад, для задач прогнозування захворюваності можуть бути застосовані регресійні моделі (Linear Regression, Random Forest Regressor, XGBoost), рекурентні нейронні мережі (LSTM, GRU), а також ансамблеві методи. Класифікаційні алгоритми (наприклад, Decision Trees, Support Vector Machines, Neural Networks) можуть бути використані для виявлення епідеміологічних ризиків або класифікації регіонів за рівнем загрози.

Особливе значення має інтеграція просторово-часових даних (spatio-temporal data), що дозволяє моделювати динаміку поширення захворювань у реальному часі з урахуванням географічних та соціальних факторів. Завдяки методам кластеризації (наприклад, DBSCAN, K-Means) можливе виявлення осередків захворювання, а застосування алгоритмів глибокого навчання дозволяє враховувати взаємозалежності між змінними, які складно формалізувати традиційними способами.

Таким чином, поєднання інформаційних технологій та методів машинного навчання відкриває нові можливості для автоматизованого епідеміологічного аналізу та раннього попередження про можливі спалахи інфекційних хвороб. Це, своєю чергою, сприяє прийняттю обґрунтованих рішень на рівні медичних установ, місцевих органів влади та систем громадського здоров'я в цілому.

1.2 Огляд та аналіз існуючих рішень

Прогнозування захворюваності є важливим завданням у сфері епідеміології, охорони здоров'я та медичної статистики. Серед традиційних підходів до цієї проблеми широко застосовуються статистичні моделі, які дозволяють аналізувати динаміку поширення хвороб та робити прогнози на основі історичних даних [1]. До основних традиційних статистичних методів прогнозування [2] належать: моделі часових рядів, регресійні моделі, дисперсійний аналіз, дискретні моделі, Баєсівські методи.

Нижче описано два із них, а саме моделі часових рядів і регресійні моделі.

Методи аналізу часових рядів використовуються для прогнозування даних, які змінюються в часі. Вони дозволяють враховувати такі особливості, як сезонність, тренди та циклічні коливання. Це є важливим при прогнозуванні інфекційних захворювань. Серед найбільш поширених моделей такого типу є:

- ARIMA [3] - авторегресійна інтегрована модель ковзного середнього. Ця модель використовується для прогнозування залежностей у часових рядах, дозволяючи враховувати вплив попередніх значень на майбутні. Вона добре працює для стаціонарних процесів, де немає різких змін у динаміці захворюваності;
- SARIMA [4] - розширення ARIMA, яке включає сезонні компоненти. Ця модель особливо корисна для прогнозування інфекційних захворювань, що мають виражену сезонну динаміку, таких як грип або ГРВІ;
- методи експоненціального згладжування [5] - підхід, який використовує згладжування з ваговими коефіцієнтами, що експоненційно зменшуються для

давніших спостережень. Це дозволяє швидше реагувати на останні зміни в даних і є корисним для короткострокового прогнозування.

Регресійний аналіз дозволяє моделювати залежність між рівнем захворюваності та іншими факторами, такими як погодні умови, рівень вакцинації, соціальні фактори тощо. Деякі основні підходи:

- лінійна регресія [6] - використовується для оцінки залежностей між захворюваністю та зовнішніми факторами, проте має обмеження при моделюванні складних нелінійних процесів;
- множинна регресія [7] - дозволяє враховувати декілька незалежних змінних, що впливають на рівень захворюваності;
- логістична регресія [8] - часто застосовується для моделювання ймовірності виникнення певної події, наприклад, спалаху інфекції в залежності від певних умов.

Стикаючись із задачею прогнозування, використовуючи такі моделі, різні дослідники пропонують різні підходи до її вирішення. Кожен з яких дає рішення згадуваної задачі, хоч і має недоліки та обмеження, на які необхідно звернути увагу для досягнення кращих результатів.

Наприклад сучасні дослідження демонструють ефективність методів машинного навчання для прогнозування пандемії COVID-19. Зокрема, LSTM-мережі показали високу точність у передбаченні кількості нових випадків зараження. Дослідження Chimmula & Zhang (2020) [13] підтвердило, що LSTM здатна враховувати довгострокові залежності в даних, що особливо важливо для прогнозування динаміки пандемії. Також було проведено комплексне порівняння методів [14], де ARIMA та Random Forest показали кращі результати для короткострокових прогнозів, тоді як SVR виявився ефективним для виявлення нелінійних залежностей. Ці підходи можуть бути адаптовані для інших регіонів, проте їх точність залежить від якості та обсягу вхідних даних.

Робота [9] являє собою глибокий огляд потенційної ролі машинного навчання у прогнозуванні поширення інфекційних захворювань. Дана робота

відкриває широкі можливості для подальших досліджень у галузі медичного прогнозування.

У цій роботі автори класифікували вже існуючі моделі за такими категоріями: статистичне прогнозування та методи на основі машинного навчання. Це дозволяє бачити різноманітність підходів та вибирати найбільш оптимальні моделі для конкретних умов застосування. Автори статті досить приділяють аналізу проблем обробки великих даних, формулювання завдання та оцінці ефективності моделей.

Однак, незважаючи на це, було виявлено низку питань, що потребують додаткової уваги з боку дослідників. Тож основним недоліком їх роботи є оглядовий характер самого дослідження. Даний огляд не надає детальних рекомендацій щодо впровадження конкретних моделей у практику, що є його слабкою стороною. Тобто замість глибокого аналізу конкретних моделей або їх практичного тестування робота дає лише поверхневі результати. Це свідчить про необхідність проведення додаткових експериментальних досліджень для уточнення ефективності запропонованих підходів.

У іншій роботі [10] увага зосереджується на розробці алгоритмів стійкого навчання для раннього виявлення ризику інфекційних захворювань. Автори приділяють особливу увагу розробці моделей, що здатні адаптуватися до змін умов середовища, що є важливим аспектом сучасних досліджень. Вони представили адаптивну модель з механізмом обрізання втрат, яка об'єднує класичні ознаки взаємної ентропії з показниками середнього коливання, що сприяє підвищенню стабільності системи при впливі шумових даних. Цей підхід дозволяє значно зменшити вплив аномальних даних на результати прогнозування. Також для зменшення впливу шуму на результати навчання було розроблено метод аналізу коефіцієнта вибірки.

Основним недоліком роботи є те, що автори зосередились на технічних аспектах надання стійкості моделі, але не провели аналіз її ефективності у реальних умовах. Таким чином, робота потребує подальшої верифікації у реальних експериментальних сценаріях для підтвердження її практичної застосовності.

У [11] розглядається застосування методів машинного та глибокого навчання для прогнозування серцево-судинних захворювань на основі клінічних даних. Автори пропонують гібридну модель, що об'єднує класичні алгоритми машинного навчання із двома моделями глибокого навчання. Комбінація різних підходів дозволяє досягти більшої точності в прогнозуванні, що є досить важливим для клінічної практики. Були отримані високі показники точності, що були досягнуто за допомогою модифікованих алгоритмів визначення вагових коефіцієнтів. Це свідчить про ефективність обраних методик та можливість їх подальшого впровадження у системи підтримки прийняття рішень.

Робота [12] пропонує систему раннього попередження про епідемічні спалахи шляхом інтеграції різноманітних джерел даних, у тому числі історичних медичних показників, інформації з соціальних мереж та метеорологічних даних. Підхід, запропонований авторами, дозволяє створити комплексну систему, яка враховує багато чинників, що впливають на поширення епідемій.

У цій роботі автори довели, що завдяки комбінуванню методів класичного машинного та глибокого навчання можна створити модель, яка своєчасно може виявити потенційну загрозу (епідемію). При цьому є висока залежність від якості вхідних даних та обмежена адаптивність до певних регіональних особливостей, через що модель буде потребувати калібрування для кожного конкретного випадку. Це підкреслює необхідність розробки універсальних алгоритмів, здатних адаптуватися до різноманітних умов без значного втручання людини.

Завдяки аналізу існуючих рішень було виявлено що методи машинного та глибокого навчання вже були застосовані для вирішення задачі прогнозування інфекційних захворювань і непоганий результат тому використання таких методів має потенціал для використання в прогнозуванні інфекційних захворювань. Проте у вищезгаданих роботах були певні недоліки та обмеження, які не дозволяють зробити висновок, що задача прогнозування епідеміологічних ситуацій є вирішеною.

Також важливо зазначити, що програмна реалізація має бути практичною та здатною швидко пристосовуватись до змін у вхідних даних.

1.3 Постановка задачі

На основі аналізу існуючих рішень та виявлених обмежень у сфері прогнозування інфекційних захворювань було сформульовано основні завдання, що необхідно вирішити для створення сучасної системи прогнозування з використанням методів машинного навчання. Сучасні підходи демонструють високий потенціал, проте мають ряд недоліків, серед яких:

- висока залежність від вхідних даних;
- обмежена адаптивність моделей до регіональних особливостей та змін умов середовища;
- відсутність практичних рекомендацій щодо впровадження розроблених алгоритмів у реальних умовах.

З огляду на виявлені проблеми, основними завданнями є:

- створення надійної системи збору та обробки даних: забезпечення автоматизованого збору даних із державних порталів, медичних установ, соціальних мереж та інших джерел з подальшим їх очищенням, нормалізацією та інтеграцією. Це дозволить отримати репрезентативну базу даних для подальшого аналізу;
- створення адаптивної моделі прогнозування: розробка алгоритмів, що використовують методи машинного та глибокого навчання для побудови прогнозової моделі. Особлива увага має бути приділена адаптації моделі до змін у вхідних даних та регіональних особливостей, що дозволить мінімізувати вплив аномальних даних та шуму;
- валідація та оптимізація моделі: забезпечення постійного контролю якості прогнозів за допомогою таких показників, як MAE, RMSE тощо, з метою оптимізації параметрів моделі та підвищення її точності;
- розробка засобів візуалізації: створення зручного інтерфейсу та візуалізації результатів у вигляді дашбордів, графіків та звітів дозволить оперативно аналізувати динаміку захворюваності та приймати своєчасні управлінські рішення;

– підвищення практичної орієнтованості розробленого рішення: забезпечення адаптивності та масштабованості системи, що дозволить використовувати її в умовах різних регіонів і сценаріїв поширення інфекційних захворювань без значного втручання користувача.

Запропоноване рішення дозволить не лише зменшити час обробки даних та підвищити точність прогнозування, а й сприятиме покращенню профілактичних заходів за рахунок своєчасного виявлення потенційних епідемічних загроз.

Використання проєктованого модуля прогнозування сприятиме ефективнішому прийняттю управлінських рішень у сфері охорони здоров'я та забезпечить надійну інформаційну підтримку для реагування на змінні епідеміологічні умови.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ

2.1 Загальна структура проектного модулю аналізу та прогнозування інфекційних захворювань

Для ефективної реалізації процесу прогнозування інфекційних захворювань необхідно розробити модуль із чітко структурованою архітектурою, що охоплює всі етапи роботи з даними - від їх збору до побудови та використання прогнозної моделі. Загальна структура модуля формується на основі функціонального поділу цього процесу на транзакційну та аналітичну складові, що забезпечує логічну послідовність обробки інформації та її використання для прийняття рішень.

Під час проектування було враховано вимоги до масштабованості, адаптивності та інтеграції із зовнішніми джерелами даних. Основна увага приділяється розмежуванню функцій збору, трансформації, аналізу та візуалізації даних. Завдяки цьому структура модуля дозволяє гнучко реагувати на зміну зовнішніх умов, швидко оновлювати інформацію та формувати точні прогнози на основі актуальних даних.

Архітектура модуля побудована з урахуванням можливості багатоетапної обробки, включаючи попередню фільтрацію, нормалізацію та трансформацію даних. Окремий блок відведено для машинного навчання, де реалізуються алгоритми побудови прогнозних моделей з можливістю автоматичного налаштування параметрів. Особливе значення має також візуалізація результатів - вона виконує роль аналітичного інструмента, що дозволяє виявити ключові закономірності та підтримати прийняття управлінських рішень.

Таким чином, загальна структура проектного модуля являє собою послідовну й логічно пов'язану систему компонентів, кожен з яких виконує окрему функціональну роль у забезпеченні надійного прогнозування епідеміологічної ситуації.

Для наочного представлення загальної структури проектного модуля прогнозування захворюваності інфекційними хворобами доцільно використати схематичне зображення ключових інформаційних потоків і компонентів системи.

На рисунку 2.1 представлено діаграму, де продемонстровані основні вхідні та вихідні дані, а також технічні й організаційні ресурси необхідні для роботи модуля.



Рисунок 2.1 - IDEF0 для модуля прогнозування

Центральним елементом є блок «Прогнозування захворюваності інфекційних хвороб», який виконує основну обчислювальну функцію - обробку вхідних даних, побудову прогнозової моделі та генерацію результатів. Вхід до модуля забезпечується декількома інформаційними та технічними потоками. До інформаційних джерел відносяться:

- а) історичні дані про захворюваність, які є базою для навчання моделей;
- б) демографічні дані, що дозволяють враховувати регіональні особливості населення;
- в) регуляторні вимоги та метрики якості, які визначають точність і достовірність прогнозів;

г) стандарти обробки даних та умови користування, що регламентують доступ і формат даних;

д) гіперпараметри, які використовуються для налаштування моделей машинного навчання.

Технічна підтримка включає:

а) алгоритми машинного навчання, що є ядром розрахунків;

б) програмний код, який реалізує логіку системи;

в) алгоритми зчитування даних, що забезпечують інтеграцію із зовнішніми джерелами;

г) персонал, що виконує технічне та аналітичне обслуговування.

На виході модуль формує звітні дані, які подаються у вигляді узагальненої статистики, прогнозів або графічних звітів. Ці результати використовуються для подальшого аналізу та прийняття управлінських рішень у сфері громадського здоров'я.

Таким чином, діаграма демонструє цілісну архітектуру взаємозв'язаних компонентів, що забезпечують ефективну роботу системи прогнозування у складному середовищі з великою кількістю вхідних параметрів та обмежень.

Для більш детального розуміння функціонування системи прогнозування захворюваності інфекційними хворобами, на діаграмі на рисунку 2.2 зображено послідовний процес обробки інформації - від первинного зчитування даних до формування кінцевих звітів. Кожен етап представлено як окремий блок із вхідними та вихідними потоками даних, що дозволяє чітко простежити логіку системи.

Процес розпочинається зі зчитування інформації, де використовуються історичні дані про захворюваність та демографічні дані. Цей етап також враховує умови користування даними та алгоритми зчитування, що забезпечують коректне отримання необхідної інформації.

Далі дані передаються до блоку обробки, де з урахуванням стандартів обробки, а також з використанням програмного коду, інформація приводиться до уніфікованого формату, очищується та нормалізується. Результатом є оброблені дані, які можуть бути використані для моделювання.

Третім етапом є створення прогнозів, яке виконується із застосуванням алгоритмів машинного навчання, налаштованих за допомогою гіперпараметрів, а також із урахуванням метрик якості, які контролюють точність моделі. У цьому ж блоці враховуються регуляторні вимоги, що визначають допустимі межі похибок і формат вихідних даних.

На фінальному етапі - створення звітів - модуль формує звітні аналітичні дані, що можуть включати графіки, таблиці, аналітичні висновки тощо. Тут беруть участь персонал і програмний код, які забезпечують правильне представлення результатів у потрібному форматі.

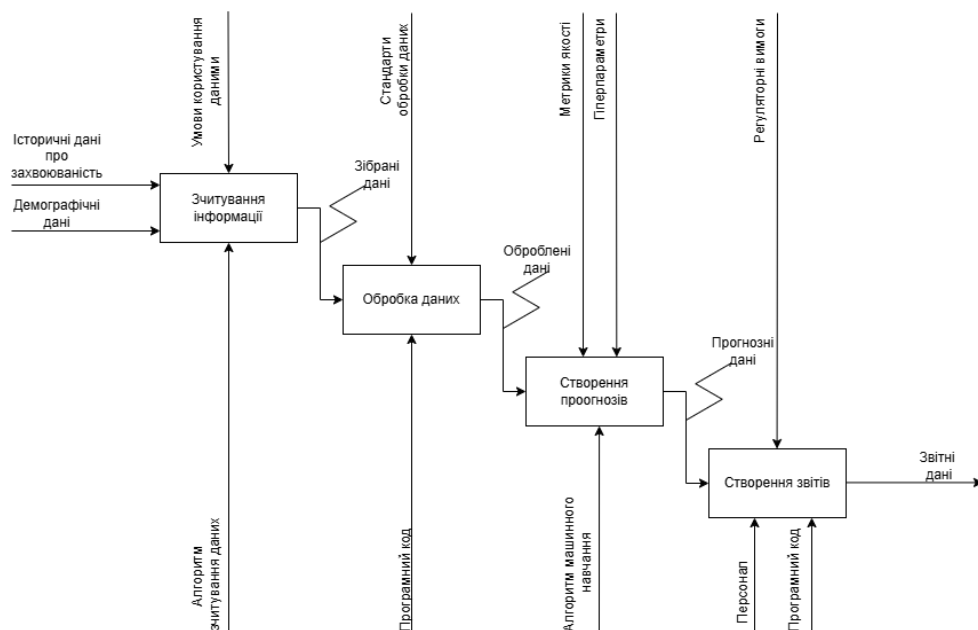


Рисунок 2.2 - Діаграма потоку даних для модуля прогнозування

Доцільно зазначити, що вхідні дані мають бути добре структуровані і метод машинного навчання для прогнозування має бути вибрано оптимально.

2.2 Алгоритмічне забезпечення модулю аналізу та прогнозування інфекційних захворювань

Для розуміння принципу роботи модуля прогнозування захворюваності інфекційними хворобами було розроблено поетапний алгоритм, що забезпечує

послідовну обробку даних, генерацію прогнозів і виведення результатів. Наведена на рисунку 2.3 схема алгоритму демонструє логіку функціонування системи на всіх основних етапах - від початкового зчитування даних до отримання підсумкового звіту.



Рисунок 2.3 - Схема алгоритму програмного модуля

Алгоритм починається з етапу зчитування даних, де здійснюється початковий збір вхідної інформації, зокрема історичних та демографічних даних.

Наступним кроком є перевірка коректності отриманих даних - у випадку виявлення помилок або невідповідностей система повертається до попереднього етапу для повторного зчитування. Після успішної валідації відбувається попередня обробка даних, що включає очищення, нормалізацію та приведення інформації до єдиного формату.

Після цього система переходить до головного етапу - прогнозування, яке базується на використанні алгоритмів машинного навчання. Згенеровані прогнози проходять перевірку точності: якщо результати не задовольняють задані критерії, прогноз може бути перераховано. У фінальній частині виконується генерація звіту та виведення результатів, які можуть бути використані для аналітики, прийняття рішень або публічного інформування.

Ця схема демонструє циклічну природу процесу та контроль якості на всіх критичних точках, що є важливим фактором для забезпечення надійності та достовірності прогнозної моделі.

2.3 Метрики оцінки точності прогнозувальних моделей

У процесі розробки модуля прогнозування для визначення тенденцій розвитку певних показників, важливо не лише правильно обрати алгоритми машинного навчання, згадуваних у підрозділі 1.2, але й оцінити ефективність побудованих моделей. Оцінка точності є важливим етапом процесу моделювання, оскільки дозволяє визначити, наскільки точно модель здатна передбачати майбутні значення на основі наявних даних.

Для оцінки точності прогнозування за допомогою застосовуваних алгоритмів машинного навчання використовуються кілька стандартних метрик [18], серед яких:

- середньоквадратична похибка (RMSE - Root Mean Square Error)[15];
- середня абсолютна похибка (MAE - Mean Absolute Error)[16];
- середня абсолютна відсоткова помилка (MAPE - Mean Absolute Percentage Error)[17].

RMSE є однією з найбільш поширених метрик для оцінки точності прогностичних моделей. Вона вимірює середнє квадратичне відхилення між прогнозованими значеннями та реальними даними. Чим менший цей показник, тим точніше модель:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (2.1)$$

де y_i - реальні значення;

$\hat{y}_i$ - прогнозовані значення;

n - кількість спостережень.

MAE менш чутлива до великих помилок, ніж RMSE, і дозволяє краще оцінювати модель, яка здійснює помилки невеликої амплітуди. Ця метрика вимірює середнє абсолютне відхилення прогнозованих значень від реальних:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (2.2)$$

MAPE вимірює відсоткову різницю між реальними і прогнозованими значеннями, що робить її зручною для порівняння ефективності різних моделей на різних наборах даних. Цей показник корисний, коли потрібно оцінити, на скільки відсотків відхиляються прогнози від фактичних значень:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100, \quad (2.3)$$

Ці методи оцінки точності дають змогу не тільки побудувати точні прогнози, але й зрозуміти, на яких етапах модель потребує покращення або оптимізації. Ретельний аналіз таких метрик, а також наявність візуалізацій допомагають детально оцінити якість моделі та зробити необхідні корективи для підвищення її ефективності.

2.4 Інформаційне забезпечення модулю аналізу та прогнозування інфекційних захворювань

Інформаційне забезпечення є важливим елементом в роботі модуля, а саме процесі прогнозування. Саме якість, повнота та актуальність вхідних даних безпосередньо впливають на точність та ефективність прогнозової моделі. У даному контексті під інформаційним забезпеченням розуміється сукупність даних, джерел їх отримання, форматів подання, а також способів їх попередньої обробки та інтеграції в загальну систему.

Основними джерелами інформації є відкриті реєстри МОЗ[19]. Це забезпечує впевненість у тому, що всі дані є достовірні та точні. Вся інформація там подана у вигляді звітів. Для швидкого збору даних був реалізований алгоритм, який забезпечує зчитування потрібних даних із наданих у широкий доступ звітів та запису їх у базу даних.

Особливу увагу було приділено форматам подання інформації - дані повинні бути стандартизовані, структуровані, із чітко визначеними метаданими. Для уникнення помилок при обробці інформації впроваджуються механізми валідації та перевірки коректності на етапі збору та первинної обробки. Крім того, з метою підтримки актуальності моделі, було забезпечено регулярне оновлення даних відповідно до встановлених інтервалів.

На рисунку 2.4 представлена діаграма інформаційного забезпечення, що відображає структуру бази даних для системи прогнозування захворювань. Вона охоплює ключові сутності предметної області, їх атрибути та логічні зв'язки між таблицями. Дана модель дозволяє зберігати історичні дані щодо захворюваності, проводити аналітику, а також формувати прогнозні значення на основі раніше зафіксованих показників.

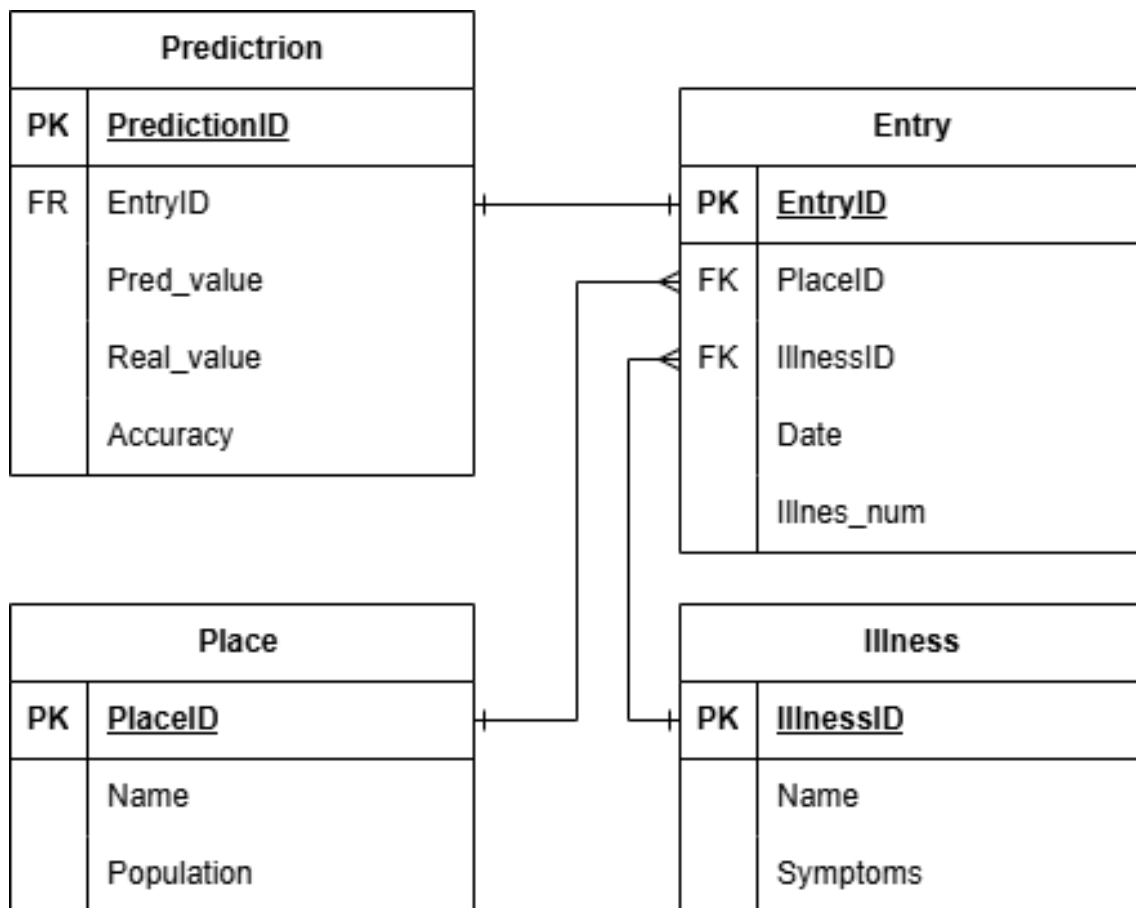


Рисунок 2.4 - ER-діаграма бази даних модуля

Основними сутностями бази даних є:

- Entry - таблиця, що відіграє роль центрального вузла у структурі. Вона містить записи про кількість випадків певного захворювання у конкретному населеному пункті на певну дату. Атрибути: EntryID (первинний ключ), PlaceID (зовнішній ключ до таблиці Place), IllnessID (зовнішній ключ до таблиці Illness), Date (дата запису), Illnes_num (кількість випадків захворювання);
- Place - таблиця, що містить інформацію про населені пункти, в яких здійснюється моніторинг. Містить поля PlaceID (первинний ключ), Name (назва населеного пункту), Population (кількість населення);
- Illness - таблиця захворювань, що зберігає довідкову інформацію про можливі хвороби. Складається з атрибутів IllnessID (первинний ключ), Name (назва захворювання), Symptoms (перелік симптомів);
- Prediction - таблиця для зберігання результатів прогнозування. Вона містить дані про передбачене значення кількості випадків, реальне значення, а

також точність прогнозу. Атрибути: PredictionID (первинний ключ), EntryID (зовнішній ключ до таблиці Entry), Pred_value (прогнозоване значення), Real_value (фактичне значення), Accuracy (точність прогнозу).

Кожна таблиця зв'язана з іншими через зовнішні ключі, що забезпечує цілісність даних і логічну зв'язаність системи. Наприклад, таблиця Prediction напряду пов'язана з Entry, яка в свою чергу поєднує інформацію про захворювання та географічну локацію. Така структура бази даних є зручною для подальшого аналізу, побудови моделей машинного навчання та формування звітів.

Представлена структура бази даних є нормалізованою [20], що дозволяє мінімізувати надлишковість даних та уникнути логічних аномалій при оновленні, вставці або видаленні записів. Дані чітко розділені між сутностями відповідно до їх функціонального призначення: окремо зберігаються населені пункти, типи захворювань, записи статистики та результати прогнозів. Це забезпечує зручність у масштабуванні системи, спрощує підтримку та оновлення інформації, а також підвищує ефективність виконання запитів до бази. Завдяки цьому система може працювати стабільно навіть при збільшенні обсягів інформації або розширенні функціональності.

Зображена на рисунку 2.5 схема ілюструє узагальнений процес роботи інформаційної системи прогнозування захворюваності, починаючи з джерел даних і закінчуючи формуванням звітної документації.

На верхньому рівні вказано основне джерело інформації - відкриті державні реєстри. Саме з них здійснюється завантаження структурованих даних до трьох основних таблиць бази даних: Place (інформація про населені пункти), Entry (записи про захворюваність) та Illness (перелік і опис захворювань).

Далі відбувається етап попередньої обробки даних, що включає в себе очищення, нормалізацію, верифікацію і, за потреби, агрегацію записів. Після цього очищені дані передаються до модуля прогнозування, який використовує таблицю Entry як основу для аналітичних розрахунків. У процесі прогнозування формується нова таблиця Prediction, яка містить очікувані значення, реальні значення (якщо вже відомі) та точність прогнозу.

Завершальним етапом є формування звітної документації - зокрема узагальнених звітів для аналітиків, статистичних оглядів чи візуалізацій, які ґрунтуються на поєднанні даних з таблиць Entry та Prediction. Така схема демонструє чіткий поділ на етапи збору, обробки, аналітики та представлення інформації, забезпечуючи логічну послідовність і цілісність процесу.

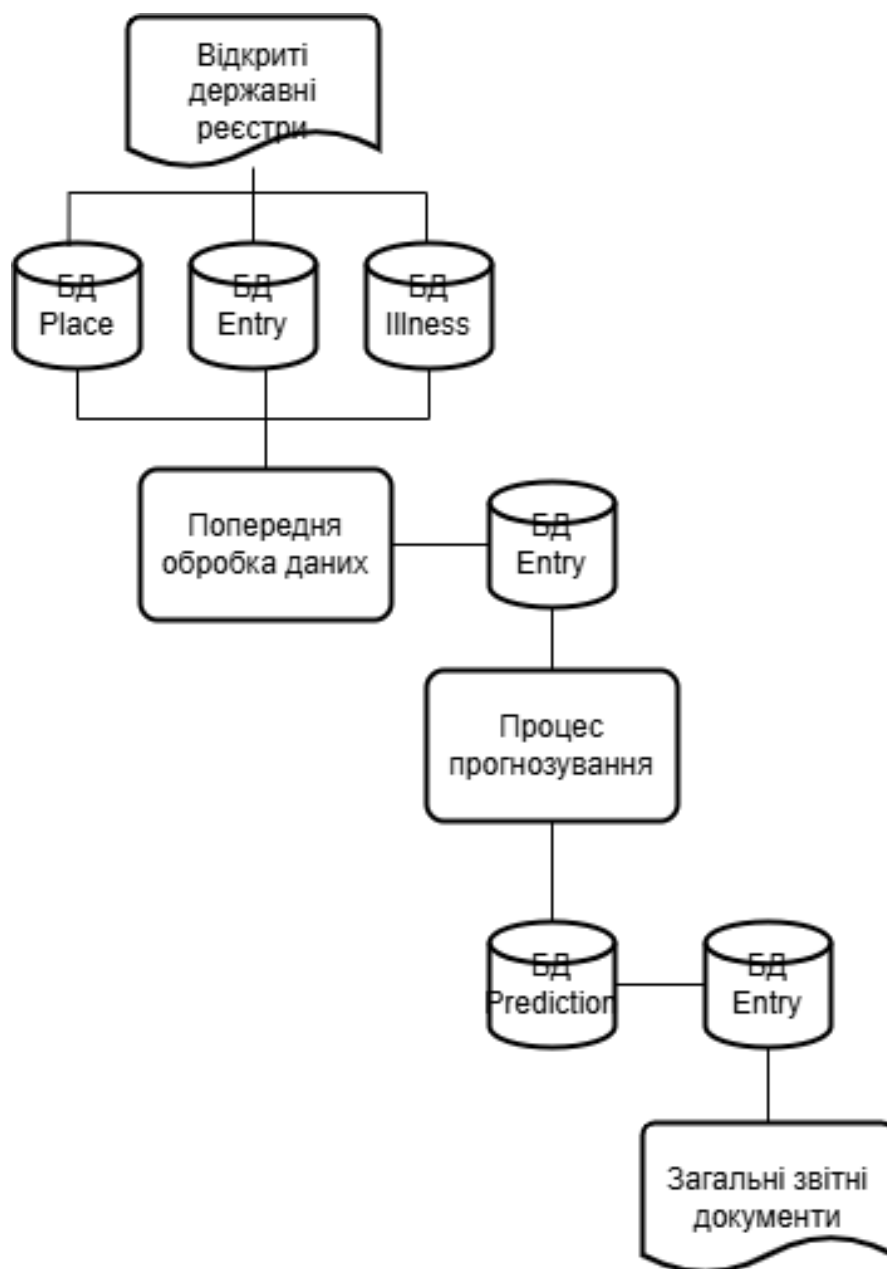


Рисунок 2.5 - Схема даних

Базу даних для реалізації даної інформаційної системи створено в системі управління базами даних MySQL [21]. Вибір саме цієї СУБД зумовлений її широким розповсюдженням, стабільністю роботи, високою продуктивністю та

підтримкою стандарту SQL. MySQL добре інтегрується з різними мовами програмування та середовищами розробки, що робить її зручною для побудови веб-орієнтованих або клієнт-серверних інформаційних систем. Обрана структура бази даних повністю відповідає можливостям MySQL, зокрема в частині підтримки зв'язків між таблицями, індексації та обробки запитів великого обсягу.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ

3.1 Реалізація програмного забезпечення модулю аналізу та прогнозування інфекційних захворювань

Для реалізації модуля було обрано мову програмування Python [22], що зумовлено її широким застосуванням у галузі машинного навчання та аналізу даних. Python має багатий набір бібліотек та інструментів, що значно спрощують процес розробки та дозволяють ефективно працювати з великими обсягами даних.

Зокрема, бібліотеки такі як Pandas, NumPy, Scikit-learn та Matplotlib є основними інструментами для обробки, аналізу та візуалізації даних, а також для реалізації алгоритмів машинного навчання. Python також є універсальним інструментом для реалізації різноманітних моделей, що є важливим аспектом для порівняння ефективності різних підходів прогнозування.

Спроектовану у підрозділі 2.3 базу даних було реалізовано у середовищі MySQL для зберігання історичних даних про захворювання, результатів прогнозування та супутньої інформації. Для взаємодії з базою даних використовувалися SQL-запити, що реалізовувалися у середовищі Python за допомогою бібліотеки mysql-connector-python.

Першим етапом роботи з даними була їх початкова обробка, яка включала нормалізацію та перевірку даних. На цьому етапі здійснювалась перевірка на наявність пропусків та аномальних значень, що могли б вплинути на якість моделювання. Для цього було використано різні методи заповнення пропусків, зокрема використання середнього значення або лінійної інтерполяції, залежно від характеру даних.

Також проводилась нормалізація значень, що дозволила привести всі показники до спільного масштабу, що є важливим для подальшої побудови моделей прогнозування. Це дозволяє уникнути дисбалансу між різними змінними та забезпечити коректне навчання моделей.

Крім того, на цьому етапі було здійснено перевірку на наявність трендів або сезонних коливань у даних, що допомогло визначити відповідні моделі для прогнозування та підготувати дані до наступних етапів аналізу.

Далі було проведено аналіз та тестування різних методів прогнозування з метою визначення найбільш ефективного підходу до прогнозування інфекційних захворювань. Було розглянуто декілька класичних та сучасних моделей, які використовуються в часових рядах, зокрема моделі SARIMA, ARIMA, Exponential Smoothing, Prophet, а також базові підходи, такі як Moving Average, Naive Forecast, Simple Average, Weighted Average. Окрім того, було реалізовано ансамблевий метод Stacking, який поєднує результати кількох моделей для підвищення точності прогнозування.

Кожну з моделей було налаштовано відповідно до специфіки наявного часового ряду, який містив щомісячні значення показника захворюваності. Усі моделі проходили процес навчання на частині даних, після чого оцінювалась точність прогнозу на відкладеній тестовій вибірці. Для порівняння якості прогнозування використовувались стандартні метрики похибки, такі як середньоквадратична похибка та середня абсолютна похибка, результати оцінки продемонстровані у таблиці 3.1.

Таблиця 3.1 - Оцінка методів прогнозування

Назва методу прогнозування	Назва метрики точності	
	RMSE	MAPE
ARIMA	7287.09	5.06%
SARIMA	7849.52	4.65 %
Exponential Smoothing	6517.85	4.16 %
Prophet	6646.38	4.30 %
Moving Average	6411.08	4.06 %

Продовження таблиці 3.1

Назва методу прогнозування	Назва метрики точності	
	RMSE	MAPE
Naive Forecast	6613.27	4.20 %.
Simple Average (Ensemble)	6515.46	4.17 %
Weighted Average (Ensemble)	6493.55	4.15 %
Stacking	1427.33	0.78 %

Такий комплексний підхід дозволив оцінити переваги та недоліки кожного з методів прогнозування в контексті досліджуваного завдання. Зокрема, моделі ARIMA та Exponential Smoothing показали добрі результати на стаціонарних ділянках ряду, тоді як модель Prophet продемонструвала кращу здатність до виявлення сезонності та трендів. Метод Stacking дозволив об'єднати прогностичні властивості кількох моделей, забезпечуючи загалом найкращу точність серед усіх протестованих підходів.

На рисунку 3.1 зображено графік прогнозу, побудованого за допомогою моделі, яка продемонструвала найвищу точність серед усіх протестованих методів. Цей прогноз відображає як реальні, так і передбачені значення показника захворюваності за обраний період. Візуалізація дозволяє наочно оцінити відповідність моделі до фактичних даних, а також демонструє її здатність відтворювати загальні тенденції та коливання часового ряду. Такий результат підтверджує доцільність використання обраного методу в задачах прогнозування епідеміологічних процесів.

Далі були створені реальні прогнози на майбутнє, з урахуванням обраної моделі, яка показала найкращі результати під час тестування. Графік зображений на рисунку 3.2. Прогнозування охоплювало короткостроковий період, що

дозволило оцінити можливі варіації рівня захворюваності в найближчому майбутньому.



Рисунок 3.1 - Результат навчання Stacking моделі для ГРВІ

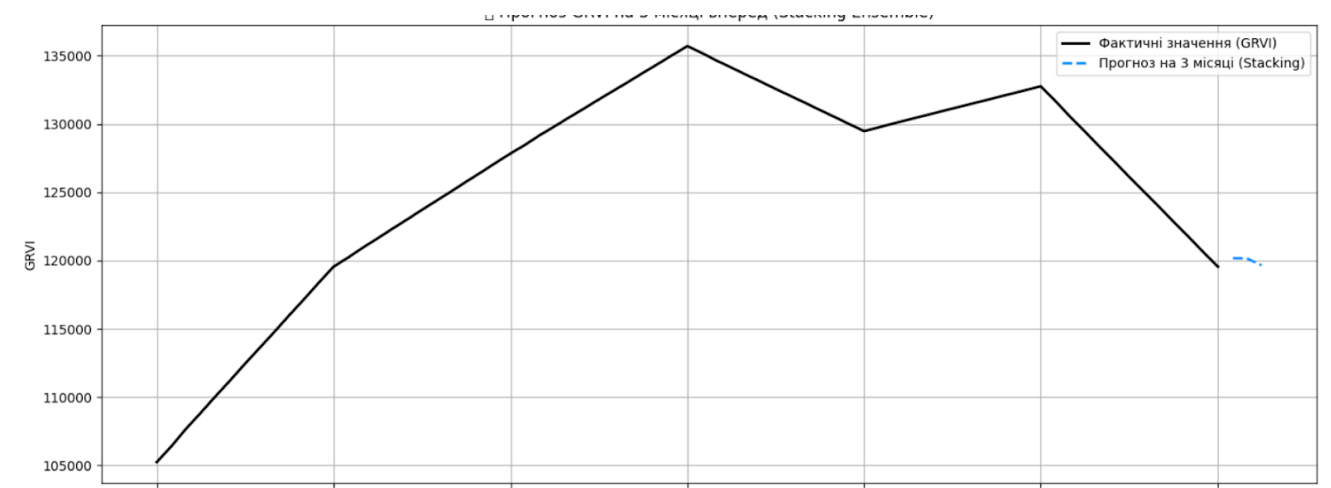


Рисунок 3.2 - Результат прогнозування Stacking моделі для ГРВІ

Для побудови графіків та візуалізації прогнозів було використано бібліотеку plotly. Це рішення було обрано через високу гнучкість та функціональність, яку надає ця бібліотека для створення інтерактивних графіків. Використання plotly дозволило не лише створити чітке та наочне представлення даних, але й забезпечити можливість взаємодії з графіками - наприклад, можливість масштабування, переміщення по осі часу та наведення на конкретні точки для отримання детальної інформації [23].

Дизайн включав оптимальне використання кольорів, чітке позначення осей, а також додаткові елементи, такі як легенда та підписи, що сприяють легкому сприйняттю інформації та аналізу прогнозів користувачем (рисунок 3.3).



Рисунок 3.3 - Графік прогнозування захворюваності створений за допомогою бібліотеки Plotly

Окрім прогнозування рівня захворюваності на грип та ГРВІ, було також здійснено побудову прогнозів для інших інфекційних захворювань, наявних у наборі даних. Для цього використовувались ті самі методи прогнозування, що й для основної мети дослідження, з урахуванням специфіки кожного захворювання та динаміки його розповсюдження. Метод Stacking також показав найкращий результат (рисунок 3.4).



Рисунок 3.4 - Графік прогнозування ВІЛ інфекції

У результаті виконаної роботи було реалізовано модуль прогнозування рівня захворюваності, що включає як побудову прогнозних моделей, так і подальшу візуалізацію отриманих результатів. Модуль працює за описаним у підрозділі 2.2 алгоритмом. Програмний код представлено у додатку А.

3.2 Інтерфейс користувача модуля аналізу та прогнозування інфекційних захворювань

Важливим аспектом є наявність зрозумілого та зручного користувацького інтерфейсу. Якісний інтерфейс значно спрощує взаємодію з аналітичною системою, дозволяючи користувачам швидко інтерпретувати результати прогнозування, перемикатися між різними регіонами, а також оперативно переглядати графіки динаміки.

Макет користувацького інтерфейсу модуля було розроблено засобами онлайн-сервісу Figma [24], який є зручним інструментом для розробки прототипів та візуалізації інтерфейсних рішень.

Макет, який представлено на рисунку 3.5, демонструє розміщення основних елементів, таких як графіки, панелі фільтрів, кнопки керування та інформаційні блоки.



Рисунок 3.5 - Макет інтерфейсу модуля у Figma

Макет дозволив чітко структурувати інтерфейс, визначити основні елементи взаємодії та стилістичне оформлення. Завдяки цьому процес верстки і подальшої інтеграції графіків відбувався швидко та узгоджено з початковою концепцією. Спроектований макет інтерфейсу модуля реалізовано за допомогою розмітки HTML та стилів CSS. Результат показаний на рисунку 3.6.

Інтерфейс було розроблено таким чином, щоб користувач міг легко обрати тип захворювання та переглянути результати прогнозу у вигляді динамічного графіка. Це сприяє кращому сприйняттю аналітичної інформації та забезпечує комфортну роботу з інструментом для різних категорій користувачів.



Рисунок 3.6 - Веб-інтерфейс модулю

Також було розроблено код на мові JavaScript, який забезпечує динамічну функціональність інтерфейсу. За допомогою JavaScript були реалізовані інтерактивні елементи, що дозволяють користувачу взаємодіяти з модулем.

Зокрема, було додано робочі кнопки для перемикання між типами захворювань, вибору регіонів, а також оновлення графіків прогнозу відповідно до вибраних параметрів. Ці елементи значно підвищують зручність використання системи, дозволяючи отримувати релевантні дані без необхідності перезавантаження сторінки.

Окрім цього, було підключено базу даних, що забезпечує оперативне оновлення інформації. Завдяки інтеграції з базою даних автоматично завантажуються актуальні дані про захворювання, що дозволяє користувачам переглядати найсвіжіші результати прогнозування без необхідності ручного оновлення.

Підключення бази даних також дало змогу реалізувати збереження історичних даних, фільтрацію за часовими періодами та регіонами, а також підтримку масштабованості системи в разі розширення обсягу аналітичної інформації.

Інтерфейс був розміщений на хостинг за допомогою сервісу HostBuddy (рисунок 3.7).

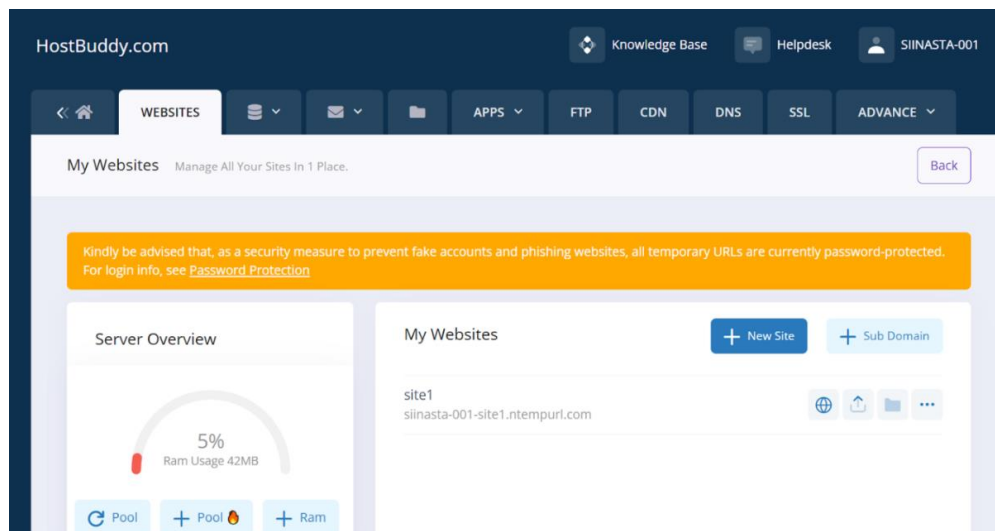


Рисунок 3.7 - Інтерфейс сервісу HostBuddy

Отже, був розроблений інтерфейс модуля та був розміщений для загального доступу. Код інтерфейсу користувача представлено у додатку Б.

3.3 Тестування розробленого модуля

Після завершення розробки інтерфейсу модуля та реалізації його функціоналу було проведено тестування., метою якого є перевірка коректності

відображення даних, працездатність інтерактивних елементів, зручність користування інтерфейсом, а також стабільність взаємодії з базою даних.

На початковому етапі тестування було виконано перевірку з'єднання між клієнтською частиною та сервером за допомогою команди ping. Це дозволило переконатися, що сервер доступний, а мережеве з'єднання встановлено правильно. Отримані відповіді свідчили про стабільну роботу мережі та готовність системи до подальшого функціонального тестування. Результат зображено на рисунку 3.8.

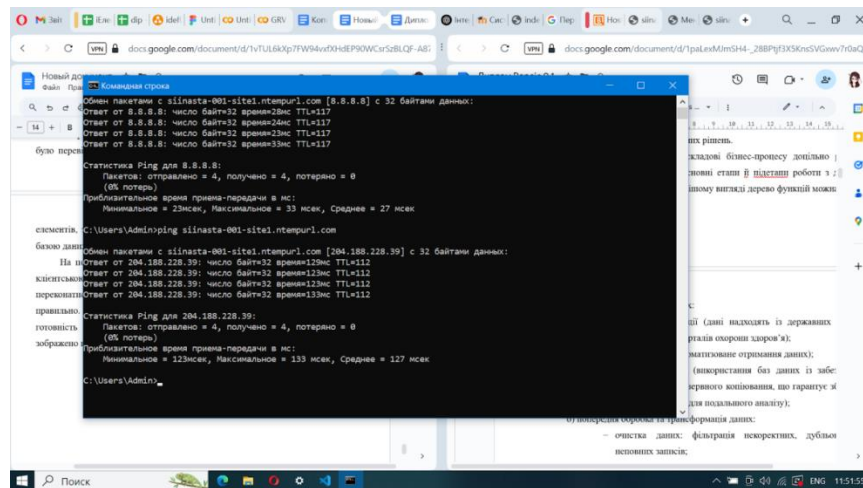


Рисунок 3.8 - Результат работы команды ping

Наступним кроком було перевірено роботу всіх інтерактивних кнопок інтерфейсу. Тестування включало перевірку реакції кнопок на наведення миші на область активації кнопки (рисунки 3.9-3.10).

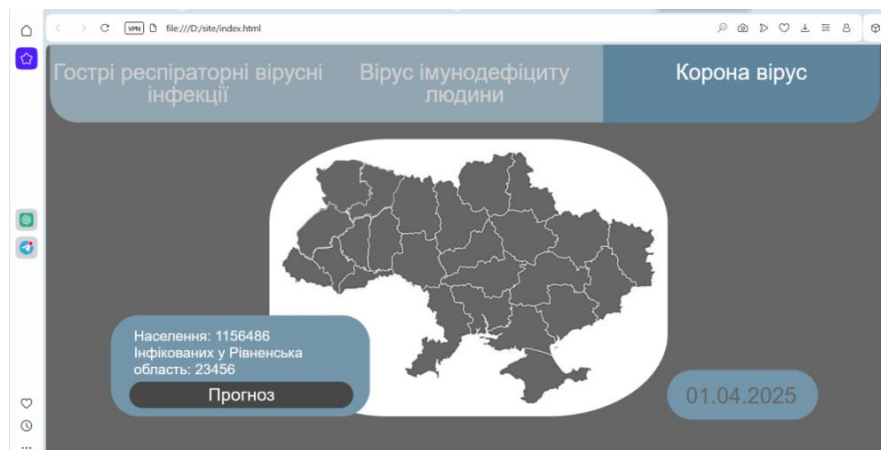


Рисунок 3.9 - Тестування реакції кнопки “Прогноз” на наведення миші

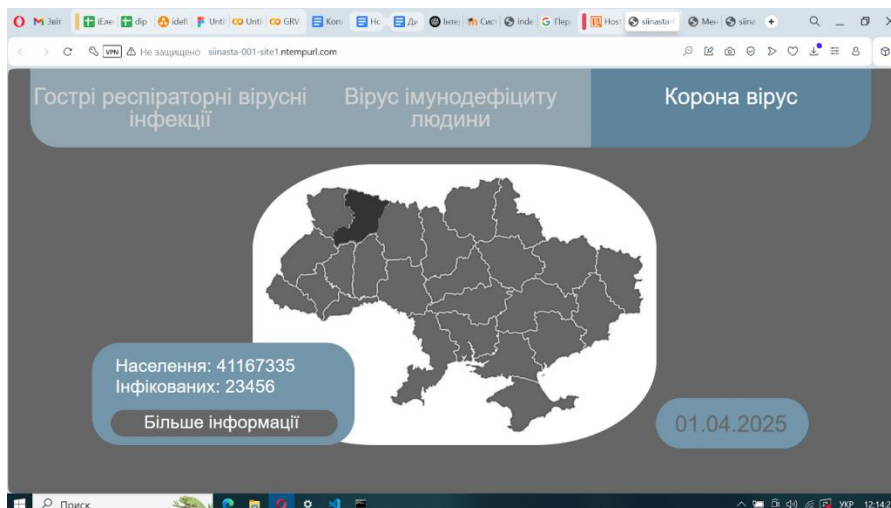


Рисунок 3.10 - Тестування реакції піктограми областей на наведення миші

Далі було перевірено роботу верхньої панелі та тригерів, пов’язаних із нею. Зокрема, тестувалась коректність відображення активної кнопки під час перемикавання між параметрами, а також оновлення інформаційного блоку, в якому відображається кількість заражених. Перевірка показала, що зміни відбуваються динамічно та узгоджено з вибраним користувачем параметром, що свідчить про правильну роботу логіки інтерфейсу. Результати зображені на рисунках 3.11 - 3.12.

На рисунку 3.11 зображено копію екрану інтерфейсу при тестуванні його верхньої панелі, а саме обрана кнопка “Гострі респіраторні вірусні інфекції”. А на рисунку 3.12 копію екрану при тестуванні верхньої панелі, а саме обрана кнопка “Вірус імунодефіциту людини”.

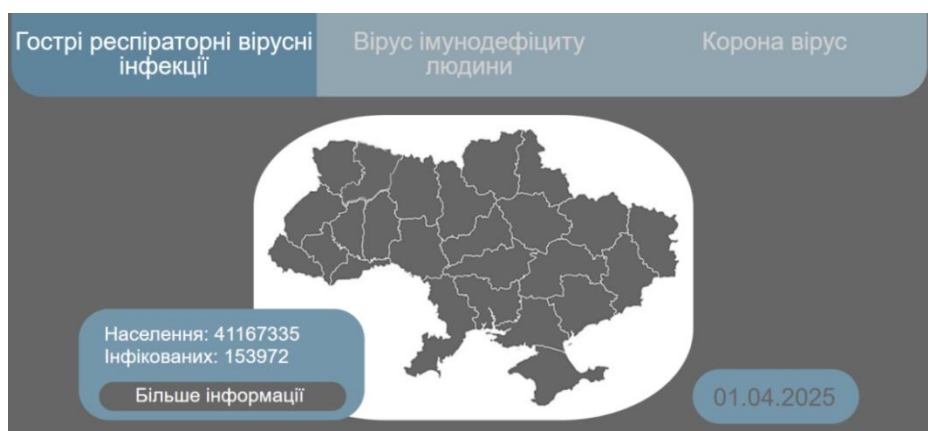


Рисунок 3.11 - Копія екрану інтерфейсу при виборі кнопки “Гострі респіраторні вірусні інфекції”



Рисунок 3.12- Тестування верхньої панелі. Обрана кнопка Вірус імунодефіциту людини

Наступним кроком було протестовано правильність відображення даних в інформаційному блоці при виборі області. Під час тестування перевірялося, чи оновлюється інформація відповідно до обраного регіону, зокрема кількість заражених та інші показники. Усі значення змінювалися коректно, що підтвердило правильну роботу зв'язку між вибором області та оновленням вмісту інформаційного блоку. Результат на рисунку 3.13.

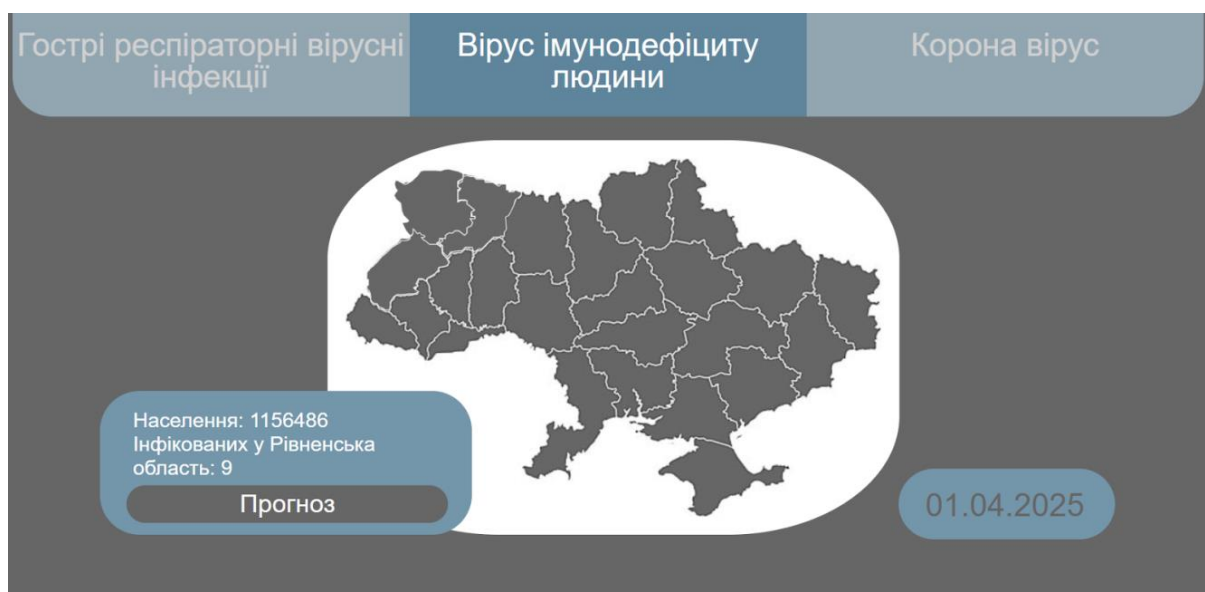


Рисунок 3.13 - Тестування правильності відображення інформації про певну область

Далі було протестовано коректність відображення графіків при натисканні на кнопку «Прогноз». Перевірка полягала у тому, щоб упевнитися, що після натискання відкривається розширений графік з деталізованими даними за вибраним параметром. Графіки відображались коректно, відповідали обраному типу захворювання та регіону (рисунок 3.14).



Рисунок 3.14 - Тестування відображення графіка

Тестування було проведено у повному обсязі, охоплюючи всі основні функціональні елементи інтерфейсу. У процесі перевірки жодних збоїв або некоректної роботи не виявлено: всі дії користувача виконувалися згідно запиту, дані оновлювалися вчасно, а всі елементи інтерфейсу функціонували відповідно до очікувань. Це дозволяє зробити висновок про готовність модуля до подальшого використання.

ВИСНОВКИ

У результаті виконаної роботи було досягнуто основної мети - створено модуль прогнозування інфекційних захворювань на основі аналізу історичних даних із використанням методів машинного навчання.

У процесі виконання кваліфікаційної роботи було:

1. Проведено аналіз предметної області та зібрано достовірні дані з відкритих державних джерел статистики захворюваності в Україні, зокрема щодо ГРВІ, ВІЛ та СНІД.

2. Проаналізовано існуючі аналоги та проведено постановку задачі.

3. Представлено загальну структуру та алгоритмічне забезпечення програмного модуля.

4. Розроблено програмний модуль прогнозування, що реалізує обрані моделі машинного навчання та забезпечує формування результатів у вигляді графіків. Для побудови візуалізацій використовувалися бібліотеки Plotly та інші інструменти в середовищі Python.

5. Створено інтерфейс користувача, який забезпечує просту взаємодію з системою. Інтерфейс має інтуїтивно зрозумілий дизайн, розроблений за допомогою Figma, що дозволяє користувачам без спеціальної підготовки переглядати та аналізувати результати прогнозування.

6. Протестовано коректну роботу програмного модуля аналізу та прогнозування інфекційних захворювань;

7. Розроблений модуль має значний потенціал для подальшого розвитку. Це може включати розширення функціональності веб-додатку, удосконалення моделей прогнозування шляхом використання ансамблевих підходів, а також інтеграцію з іншими інформаційними системами охорони здоров'я.

Розроблена система дозволяє ефективно моделювати динаміку захворюваності та представляти результати у зручному форматі для прийняття обґрунтованих рішень у галузі охорони здоров'я.

Отримані результати підтверджують ефективність застосування методів машинного навчання для моделювання епідемічних процесів і демонструють

перспективи впровадження таких рішень у практику медичних установ та органів державного управління.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Засадний, Б., Михальська, О., Кириллов, О. Використання інструментарію Business Intelligence у процесі прогнозування облікових показників підприємства. Фінансово-кредитна діяльність: проблеми теорії та практики. 2024. Т. 1, № 54. С. 244–259. DOI: <https://doi.org/10.55643/fcaptr.1.54.2024.4240>.
2. Ганчук, А. А., Соловійов, В. М., Чабаненко, Д. М. Методи прогнозування: навчальний посібник. Черкаси, 2012. URL: <https://prognoz.ck.ua/ForecastingMethodsBook-main.pdf> (дата звернення: 28.05.2025).
3. Лук'яненко, І. Г., Жук, В. М. *Аналіз часових рядів: побудова ARIMA, ARCH/GARCH моделей з використанням пакета EViews 6.0. Частина перша: практичний посібник для роботи в комп'ютерному класі*. Київ, 2013. ББК 65.261.
4. Фармага, І., Гадамський, В. Система проведення аналізу, дослідження та передбачення подій у послідовностях даних дискретного часу. Computer Design Systems. Theory and Practice. 2021. Vol. 3, No. 1. С. 61. DOI: <https://doi.org/10.23939/cds2021.01.061>. УДК 62-522.7, 004.9.
5. Ткач, Є. В. Короткотермінове прогнозування попиту методом. Вісник Західноукраїнського національного університету. Серія: Економіка. 2019. № 45. С. 123–130. [Електронний ресурс]. URL: https://library.wunu.edu.ua/images/stories/praci_vukladachiv/Факультет%20Фінансів/К%20екон.%20ан.%20стат/ткач%20є/короткотермінове%20прогнозування%20попиту%20методом.PDF (дата звернення: 28.05.2025).
6. Google Developers. Лінійна регресія // Курс з машинного навчання. 2023. [Електронний ресурс]. URL: <https://developers.google.com/machine-learning/crash-course/linear-regression?hl=uk> (дата звернення: 28.05.2025).
7. Лінійні моделі множинної регресії // Електронний посібник з фінансового менеджменту. Львів: ЛНТУ. [Електронний ресурс]. URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/Електронний%20посібник%20ФМ/page8.html (дата звернення: 28.05.2025).

8. Google Developers. Логістична регресія: функції втрат та регуляризація // Курс з машинного навчання. 2023. [Електронний ресурс]. URL: <https://developers.google.com/machine-learning/crash-course/logistic-regression/loss-regularization?hl=uk> (дата звернення: 28.05.2025).
9. Liu, M. et al. Machine Learning for Infectious Disease Risk Prediction: A Survey. arXiv. 2023. 45 p. arXiv:2308.03037 [cs.LG]. DOI: 10.48550/arXiv.2308.03037.
10. Wang, L., Chen, Y., Zhang, H. Research on Disease Prediction Model Construction Based on Computer AI Deep Learning Technology. IEEE Access. 2022. Vol. 10. P. 45321–45334. DOI: 10.1109/ACCESS.2022.3171234.
11. Alaa, A. M., Bolton, T., Di Angelantonio, E. et al. Machine Learning Methods for Predicting Cardiovascular Diseases. Nature Digital Medicine. 2019. Vol. 2, No 1. P. 1–1. DOI: 10.1038/s41746-019-0196-8.
12. Zhang, Y., Wang, H., Liu, J. Epidemic Outbreak Prediction Using Machine Learning Models. Scientific Reports. 2021. Vol. 11, No 1. Art. no. 15371. 12 p. DOI: 10.1038/s41598-021-94803-3.
13. Chimmula, V. K. R., Zhang, L. Time-series forecasting of COVID-19 transmission in Canada using LSTM networks. Chaos, Solitons & Fractals. 2020. Vol. 135. Art. no. 109864. 6 p. DOI: 10.1016/j.chaos.2020.109864.
14. Ahmad, M. W., Reynolds, J., Rezgui, Y. Time series forecasting of COVID-19 transmission in Asia Pacific countries using deep neural networks. Computers in Biology and Medicine. 2021. Vol. 129. Art. no. 104142. 14 p. DOI: 10.1016/j.combiomed.2020.104142.
15. Root Mean Square Error (RMSE) – ML Glossary // IT Wiki : [вебсайт]. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/root-mean-square-error> (дата звернення: 28.05.2025).
16. Roberts, A. Mean Absolute Percentage Error (MAPE): What You Need To Know. Arize. 2023. 02 лютого. [Електронний ресурс]. URL: <https://arize.com/blog-course/mean-absolute-percentage-error-mape-what-you-need-to-know/> (дата звернення: 28.05.2025).
17. Jierula, A., Wang, S., Oh, T.-M., Wang, P. Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural

Networks with Acoustic Emission Data. Applied Sciences. 2021. Vol. 11, No 5. Art. no. 2314. DOI: <https://doi.org/10.3390/app11052314>.

18. Центр громадського здоров'я МОЗ України. Моніторинг і статистика. [Електронний ресурс]. URL: <https://phc.org.ua/monitoring-i-statistika> (дата звернення: 28.05.2025).

19. Beeri, C., Bernstein, P. A., Goodman, N. A Sophisticate's Introduction to Database Normalization Theory. Proceedings of the 4th International Conference on Very Large Data Bases (VLDB). 1978. P. 113–124. DOI: <https://doi.org/10.1016/B978-0-934613-53-8.50035-2>. URL: <https://www.microsoft.com/en-us/research/publication/a-sophisticates-introduction-to-database-normalization-theory/> (дата звернення: 28.05.2025).

20. MySQL // Oracle Corporation. [Електронний ресурс]. URL: <https://www.mysql.com> (дата звернення: 28.05.2025).

21. Python Documentation // Python Software Foundation. [Електронний ресурс]. URL: <https://www.python.org/doc/> (дата звернення: 28.05.2025).

22. Plotly Python Documentation // Plotly Technologies Inc. [Електронний ресурс]. URL: <https://plotly.com/python/> (дата звернення: 28.05.2025).

23. Figma // Figma, Inc. [Електронний ресурс]. URL: <https://www.figma.com> (дата звернення: 28.05.2025).

24. HostBuddy Control Panel // HostBuddy. [Електронний ресурс]. URL: <https://member5.hostbuddy.com/cp/boxinfo> (дата звернення: 28.05.2025).

25. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. С.52.

26. Касьян Т. В., Турченко І. В. Програмний модуль аналізу і прогнозування інфекційних захворювань з використанням методів машинного навчання. Інтелектуальні інформаційні технології в прикладних дослідженнях: збірник тез доп. студент. наук.-практ. конф. (ІТAR-2025), м. Тернопіль, 27-29 травня 2025 р. Тернопіль: ЗУНУ, 2025. С. 160-163.

Додаток А

Програмний код процесу прогнозування

```
import pandas as pd
import numpy as np
from prophet import Prophet
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error

data = pd.read_csv('/content/dip.csv', parse_dates=['Date'])
data = data[['Date', 'GRVI']].dropna().sort_values('Date')
data = data.rename(columns={'Date': 'ds', 'GRVI': 'y'})
data = data.set_index('ds').asfreq('MS')
data['y'] = data['y'].interpolate(method='linear')
data = data.reset_index()

data['month'] = data['ds'].dt.month
data['quarter'] = data['ds'].dt.quarter

split_index = int(len(data) * 0.8)
train = data.iloc[:split_index]
test = data.iloc[split_index:]

prophet = Prophet()
prophet.fit(train[['ds', 'y']])
future = test[['ds']].copy()
forecast_prophet = prophet.predict(future)
pred_prophet = forecast_prophet['yhat'].values

es_model = ExponentialSmoothing(train['y'], seasonal='add', seasonal_periods=12)
es_fit = es_model.fit()
pred_es = es_fit.forecast(len(test))

pred_ma = np.full(len(test), train['y'].rolling(window=3).mean().iloc[-1])

valid_mask = ~np.isnan(pred_prophet) & ~np.isnan(pred_es.values) &
~np.isnan(pred_ma)
true = test['y'].values[valid_mask]

meta_X = np.column_stack([
    pred_prophet[valid_mask],
    pred_es.values[valid_mask],
    pred_ma[valid_mask],
```

```

    test['month'].values[valid_mask],
    test['quarter'].values[valid_mask]
])

meta_model = RandomForestRegressor(n_estimators=100, random_state=42)
meta_model.fit(meta_X, true)
final_pred = meta_model.predict(meta_X)

def evaluate_model(true, pred):
    mae = mean_absolute_error(true, pred)
    rmse = np.sqrt(mean_squared_error(true, pred))
    mape = np.mean(np.abs((true - pred) / true)) * 100
    return {'MAE': mae, 'RMSE': rmse, 'MAPE': mape}

metrics = evaluate_model(true, final_pred)

future_months = 3
last_date = data['ds'].max()
future_dates = pd.date_range(start=last_date + pd.DateOffset(months=1),
                             periods=future_months, freq='MS')

future_df = pd.DataFrame({'ds': future_dates})
future_df['month'] = future_df['ds'].dt.month
future_df['quarter'] = future_df['ds'].dt.quarter

prophet = Prophet()
prophet.fit(data[['ds', 'y']])
forecast_prophet = prophet.predict(future_df)
pred_prophet = forecast_prophet['yhat'].values

es_model = ExponentialSmoothing(data['y'], seasonal='add', seasonal_periods=12)
es_fit = es_model.fit()
pred_es = es_fit.forecast(future_months)

pred_ma = np.full(future_months, data['y'].rolling(window=3).mean().iloc[-1])

X_meta = np.column_stack([
    pred_prophet,
    pred_es.values,
    pred_ma,
    future_df['month'].values,
    future_df['quarter'].values
])

meta_model = RandomForestRegressor(n_estimators=100, random_state=42)

```

```

X_train = np.column_stack([
    prophet.predict(data[['ds']])['yhat'].values,
    es_fit.fittedvalues.values,
    np.full(len(data), data['y'].rolling(window=3).mean().iloc[-1]),
    data['month'].values,
    data['quarter'].values
])
y_train = data['y'].values
meta_model.fit(X_train, y_train)

forecast_stacked = meta_model.predict(X_meta)
import plotly.graph_objects as go
import numpy as np
import pandas as pd

full_ds = pd.concat([data['ds'], future_df['ds']])
full_y = np.concatenate([data['y'].values, forecast_stacked])
cutoff_date = data['ds'].max()

cut_idx = np.where(full_ds == cutoff_date)[0][0]
fig = go.Figure()

fig.add_trace(go.Scatter(
    x=full_ds[:cut_idx+1],
    y=full_y[:cut_idx+1],
    fill='tozeroy',
    mode='none',
    fillcolor='rgba(0,191,255,0.2)',
    name='Історія (заливка)'
))
fig.add_trace(go.Scatter(
    x=full_ds[cut_idx:],
    y=full_y[cut_idx:],
    fill='tozeroy',
    mode='none',
    fillcolor='rgba(255,99,71,0.2)',
    name='Прогноз (заливка)'
))

fig.add_trace(go.Scatter(
    x=full_ds[cut_idx+1:],
    y=full_y[cut_idx+1:],
    mode='lines',
    line=dict(color='deepskyblue', width=3),
    name='Історія'
))

```

```

fig.add_trace(go.Scatter(
    x=full_ds[cut_idx:],
    y=full_y[cut_idx:],
    mode='lines',
    line=dict(color='tomato', width=3, dash='dash'),
    name='Прогноз'
))

fig.add_vline(x=cutoff_date, line_width=1, line_dash='dot', line_color='lightgrey')
fig.update_layout(
    xaxis_title='Дата',
    yaxis_title='GRVI',
    template='plotly_dark',
    font=dict(size=14),
    width=1000,
    height=500,
    legend=dict(
        orientation='h',
        yanchor='bottom',
        y=-0.25,
        xanchor='center',
        x=0.5,
        bgcolor='rgba(0,0,0,0)',
        font=dict(size=12)
    )
)
fig.show()

```

Додаток Б

Код користувацького інтерфейсу

```
HTML:
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="globals.css" />
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div class="top-tabs">
      <div class="tab active" data-tab="grvi">Гострі респіраторні вірусні
інфекції</div>
      <div class="tab" data-tab="vih">Вірус імунodefіциту людини</div>
      <div class="tab" data-tab="corona">Корона вірус</div>
    </div>
    <div class="desktop">
      <div class="div">
        <div class="overlap">
          
          <div class="rectangle"></div>
          
          <div class="rectangle-2">
            <div class="overlap-group">
              <div class="rectangle-3"></div>
              <div class="text-wrapper" id="info-btn">Більше інформації</div>
            </div>
            <div class="element" id="stats"
              data-grvi="Інфікованих: 153972"
              data-vih="Інфікованих: 9"
              data-corona="Інфікованих: 23456">
              Населення: 41167335<br>
              <span id="infected">Інфікованих: 153972</span>
            </div>
            <!-- <div class="text-wrapper" id="info-btn">Більше інформації</div>
          </div>
          <div class="text-wrapper-2">01.04.2025</div>
        </div>
```

```

    </div>
</div>

<!-- Blue screen modal -->
<div class="modal" id="modal">
  <div class="modal-content">
    
    <div class="close-btn" id="close-btn"> ✖ </div>
  </div>
</div>

<script src="script.js"></script>
</body>
</html>

```

```

    CSS:
html, body {
  height: 100%;
  margin: 0;
  padding: 0;
  background: #666666;
}

.desktop {
  background-color: #666666;
  display: flex;
  flex-direction: row;
  justify-content: center;
  width: 100%;
  min-height: 100vh;
}

.desktop .div {
  background-color: #666666;
  width: 1500px;
  height: 660px;
  position: relative;
}

.desktop .overlap {
  position: absolute;
  width: 1279px;
  height: 504px;
  top: 30px; /* was 109px, now higher */
  left: 110px;
}

```

```
.desktop .map {  
  position: absolute;  
  width: 720px;  
  height: 498px;  
  top: 0; /* already at the top, keep as is */  
  left: 287px;  
  object-fit: cover;  
  border-radius: 30%; /* ← Овальна форма */  
  overflow: hidden;  
}
```

```
.desktop .rectangle {  
  width: 273px;  
  height: 89px;  
  top: 415px;  
  left: 1006px;  
  background-color: #7496aa;  
  position: absolute;  
  border-radius: 50px;  
}
```

```
.desktop .vector {  
  position: absolute;  
  width: 105px;  
  height: 105px;  
  top: 43px; /* was 43px, now higher */  
  left: 427px;  
}
```

```
.desktop .vector:hover {  
  filter: brightness(50%);  
  transition: filter 0.3s ease;  
}
```

```
.desktop .rectangle-2 {  
  width: 469px;  
  height: 181px;  
  top: 317px;  
  left: 0;  
  background-color: #7496aa;  
  position: absolute;  
  border-radius: 50px;  
}
```

```
.desktop .overlap-group {
```

```

position: absolute;
width: 417px;
height: 47px;
top: 119px;
left: 34px;
}

.desktop .rectangle-3 {
width: 405px;
height: 47px;
top: 0;
left: 0;
background-color: #666666;
position: absolute;
border-radius: 50px;
}

.desktop .text-wrapper {
width: 358px;
top: 3px;
left: 59px;
font-size: 32px;
white-space: nowrap;
position: absolute;
font-family: "Inter-Regular", Helvetica;
font-weight: 400;
color: #ffffff;
letter-spacing: 0;
line-height: normal;
}

.desktop .element {
width: 397px;
top: 22px;
left: 42px;
font-size: 32px;
position: absolute;
font-family: "Inter-Regular", Helvetica;
font-weight: 400;
color: #ffffff;
letter-spacing: 0;
line-height: normal;
}

.desktop .text-wrapper-2 {
position: absolute;

```

```
width: 217px;
top: 438px;
left: 1040px;
font-family: "Inter-Regular", Helvetica;
font-weight: 400;
color: #666666;
font-size: 40px;
letter-spacing: 0;
line-height: normal;
white-space: nowrap;
}
```

```
.desktop .group {
  position: absolute;
  width: 1279px;
  height: 486px;
  top: -377px;
  left: 110px;
  background-color: #a4c1d2;
  border-radius: 50px;
}
```

```
.desktop .overlap-2 {
  position: relative;
  width: 1206px;
  height: 486px;
  left: 73px;
  background-color: #7496aa;
  border-radius: 50px;
}
```

```
.desktop .rectangle-4 {
  width: 1125px;
  height: 486px;
  top: 0;
  left: 81px;
  background-color: #5f859c;
  position: absolute;
  border-radius: 50px;
}
```

```
.desktop .text-wrapper-3 {
  width: 802px;
  top: 398px;
  left: 274px;
  font-size: 40px;
```

```

    position: absolute;
    font-family: "Inter-Regular", Helvetica;
    font-weight: 400;
    color: #ffffff;
    letter-spacing: 0;
    line-height: normal;
}

.desktop .overlap-group:hover .rectangle-3 {
    filter: brightness(70%);
    transition: filter 0.3s ease;
    cursor: pointer;
}

/* Modal styles */
.modal {
    display: none;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(113, 121, 146, 0.9);
    justify-content: center;
    align-items: center;
    z-index: 1000;
}

.modal-content {
    position: relative;
    width: 80%;
    max-width: 800px;
    text-align: center;
}

.plot-image {
    width: 100%;
    border-radius: 20px;
}

.close-btn {
    position: absolute;
    top: 10px;
    right: 10px;
    font-size: 24px;
    color: black;
}

```

```

    cursor: pointer;
    background-color: white;
    border-radius: 50%;
    padding: 5px;
    width: 40px;
    height: 40px;
    display: flex;
    justify-content: center;
    align-items: center;
}

.open-modal-btn {
    position: fixed;
    bottom: 20px;
    right: 20px;
    padding: 10px 20px;
    font-size: 16px;
    background-color: #7496aa;
    color: white;
    border: none;
    border-radius: 10px;
    cursor: pointer;
}

/* Top tabs styles */
.top-tabs {
    display: flex;
    width: 1500px;          /* Match main container width */
    margin: 0 auto;         /* Center horizontally */
    border-radius: 0 0 50px 50px;
    overflow: hidden;
    position: relative;
    z-index: 10;
    padding: 0;
}

.tab {
    flex: 1 1 0;
    text-align: center;
    padding: 30px 0;
    font-size: 40px;
    font-family: "Inter-Regular", Helvetica, Arial, sans-serif;
    color: #fff;
    background: #7496aa;
    cursor: pointer;
    transition: background 0.3s;

```

```

border: none;
outline: none;
user-select: none;
}

.tab:not(.active) {
  opacity: 0.7;
  background: #a4c1d2;
}

.tab.active {
  background: #5f859c;
  opacity: 1;
}

.element {
  margin-bottom: 30px; /* Add space below the stats text */
  display: block;
  font-size: 40px;
  color: #fff;
  line-height: 1.2;
}

```

JavaScript:

```

document.addEventListener("DOMContentLoaded", () => {
  const modal = document.getElementById("modal");
  const infoBtn = document.getElementById("info-btn");
  const closeModalBtn = document.getElementById("close-btn");

  infoBtn.addEventListener("click", () => {
    modal.style.display = "flex";
  });

  closeModalBtn.addEventListener("click", () => {
    modal.style.display = "none";
  });

  modal.addEventListener("click", (e) => {
    if (e.target === modal) {
      modal.style.display = "none";
    }
  });
});

```

```

const tabs = document.querySelectorAll('.tab');
const infectedSpan = document.getElementById('infected');
const stats = document.getElementById('stats');
let selectedDisease = 'grvi';

tabs.forEach(tab => {
  tab.addEventListener('click', () => {
    tabs.forEach(t => t.classList.remove('active'));
    tab.classList.add('active');
    selectedDisease = tab.getAttribute('data-tab');
    const infected = stats.getAttribute('data-' + selectedDisease);
    infectedSpan.textContent = infected;
    stats.style.fontSize = "";
    autoShrinkText(stats);
  });
});

const rivne = document.querySelector('.vector');
rivne.addEventListener('click', () => {
  const rivnePatients = rivne.getAttribute('data-' + selectedDisease);
  infectedSpan.textContent = `Інфікованих у Рівненській області: ${rivnePatients}`;
  stats.style.fontSize = "";
  autoShrinkText(stats);
});

autoShrinkText(stats);
});

function autoShrinkText(element, minFontSize = 18) {
  let fontSize = parseInt(window.getComputedStyle(element).fontSize);
  while (element.scrollHeight > element.offsetHeight && fontSize > minFontSize) {
    fontSize--;
    element.style.fontSize = fontSize + "px";
  }
}

```

Додаток В
Апробація отриманих результатів

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління



ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції
ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ
ДОСЛІДЖЕННЯХ
(ІТАР-2025)

27-29 травня 2025 року

Тернопіль
2025

Карнидал Володимир, Биковий Павло	155
ПРОГРАМНИЙ МОДУЛЬ ВИЗНАЧЕННЯ ВІКУ ТА СТАТІ ЛЮДИНИ НА ЗОБРАЖЕННІ ІЗ ЗАСТОСУВАННЯМ OPENCV ТА ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ	155
Касьян Тамара, Турченко Ірина	160
ПРОГРАМНИЙ МОДУЛЬ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ	160
Кацидим Віта, Ліп'яніна-Гончаренко Христина, Ралік Ігор	164
ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ЗАДОВОЛЕНOSTІ КЛІЕНТІВ ОБСЛУГОВУВАННЯМ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ	164
Климчук Анастасія, Турченко Ірина	168
ІНТЕРАКТИВНИЙ ЗАСТОСУНОК ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ НА ОСНОВІ МЕДІА-КОНТЕНТУ ІЗ ВИКОРИСТАННЯМ OPENAI	168
Ковальковський Віталій, Ліп'яніна-Гончаренко Христина	171
МОДУЛЬ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ	171
Ковтуненко Андрій, Ліп'яніна-Гончаренко Христина	176
TELEGRAM-БОТ ДЛЯ ВІДСТЕЖЕННЯ КУРСУ КРИПТОВАЛЮТ	176
Котов Владислав, Майків Ігор	179
МОДУЛЬ КЛАСИФІКАЦІЇ АНОТАЦІЙ НАУКОВИХ СТАТЕЙ З arXiv ЗА ДОПОМОГОЮ МОДЕЛІ RoBERTa	179
Кулик Степан	182
ВИЯВЛЕННЯ МЕРЕЖЕВИХ АТАК У ТРАФІКУ ВЕЛИКИХ ДАНИХ НА ОСНОВІ ЕВОЛЮЦІЙНИХ ОБЧИСЛЕНЬ	182
Мельничук Руслана, Ліп'яніна-Гончаренко Христина	185
МОДУЛЬ ОПТИМІЗАЦІЇ МАРКЕТИНГОВИХ КАМПАНІЙ ЗА ДОПОМОГОЮ КЛАСТЕРИЗАЦІЇ КЛІЕНТІВ	185
Матвійшин Наталія, Майків Ігор	188
МОДУЛЬ АНАЛІЗУ ТА КЛАСИФІКАЦІЇ СЮЖЕТІВ ФІЛЬМІВ ЗА ЖАНРАМИ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ SPACY	188
Мукан Павло, Лендюк Тарас	191
МОДУЛЬ ВИЯВЛЕННЯ МОВИ ВОРОЖНЕЧІ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ SCIKIT-LEARN ДЛЯ КЛАСИФІКАЦІЇ ТЕКСТУ	191
Наконечна Ірина, Ліп'яніна-Гончаренко Христина	194
ВИЗНАЧЕННЯ ДЖЕРЕЛА ТЕКСТУ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖІ НА ОСНОВІ EMBEDDING І LSTM	194

Касьян Тамара
студентка групи КН-41
kasiantomochka@gmail.com

Турченко Ірина
к.т.н., доцент
itu@wunu.edu.ua

*Західноукраїнський національний університет
Тернопіль, Україна*

ПРОГРАМНИЙ МОДУЛЬ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ

У сучасному світі зростаюча кількість інфекційних захворювань ставить серйозні виклики перед системами охорони здоров'я. Прогнозування та управління епідеміологічними процесами є важливими для запобігання поширення хвороб. Використання методів машинного навчання дозволяє аналізувати історичні дані та передбачати майбутні спалахи, що сприяє оптимізації ресурсів і своєчасному вжитку заходів.

Сучасні дослідження демонструють ефективність методів машинного навчання для прогнозування інфекційних захворювань. Наприклад, моделі ARIMA та SARIMA використовуються для аналізу часових рядів із врахуванням сезонності, що є критично важливим для таких хвороб, як грип або ГРВІ [1]. Інші підходи, такі як Prophet від Facebook, дозволяють враховувати нелінійні тренди та особливості даних, що підвищує точність прогнозів [2].

Ансамблеві методи, зокрема Stacking, поєднують переваги кількох моделей, що дозволяє досягти вищої точності порівняно з окремими алгоритмами [3]. Однак існуючі рішення часто мають обмеження, такі як залежність від якості вхідних даних або відсутність адаптації до регіональних особливостей. Це підкреслює необхідність розробки більш гнучких та точних систем прогнозування.

Розроблений програмний модуль для аналізу та прогнозування інфекційних захворювань поєднує алгоритми машинного навчання з інтуїтивно зрозумілим інтерфейсом. У загальному вигляді дерево функцій модуля прогнозування інфекційних захворювань представлено на рисунку 1.



Рисунок 1 – Дерево функцій модуля прогнозування інфекційних захворювань

Інформаційне забезпечення, під яким розуміють сукупність джерел даних, форматів їх подання, а також методів попередньої обробки та інтеграції в систему, є ключовим чинником точності прогнозової моделі. Якість, повнота та актуальність вхідних даних безпосередньо впливають на ефективність роботи модуля. Основним джерелом інформації є відкриті реєстри МОЗ, що забезпечує достовірність та актуальність даних. Для автоматизації збору реалізовано алгоритм, який зчитує необхідну інформацію зі звітів і зберігає її в базі даних.

Особливу увагу приділено стандартизації форматів, структурованості даних та визначенню метаданих. Для підвищення надійності впроваджено механізми валідації та регулярне оновлення інформації згідно з заданим інтервалом.

Експерименти показали, що серед усіх протестованих моделей ансамблевий метод Stacking досяг найвищої точності з RMSE 1427.33 та MAPE 0.78% (див. таблицю 1). Для роботи методу Stacking було використано Prophet, експоненційне згладжування та ковзне середнє.

Таблиця 1 - Оцінка методів прогнозування

Назва методу прогнозування	Назва метрики точності	
	RMSE	MAPE
ARIMA	7287.09	5.06%
SARIMA	7849.52	4.65 %
Exponential Smoothing	6517.85	4.16 %
Prophet	6646.38	4.30 %
Simple Average (Ensemble)	6515.46	4.17 %
Weighted Average (Ensemble)	6493.55	4.15 %
Stacking (запропонований)	1427.33	0.78 %

На рисунку 2 представлено результати навчання моделі для ГРВІ.

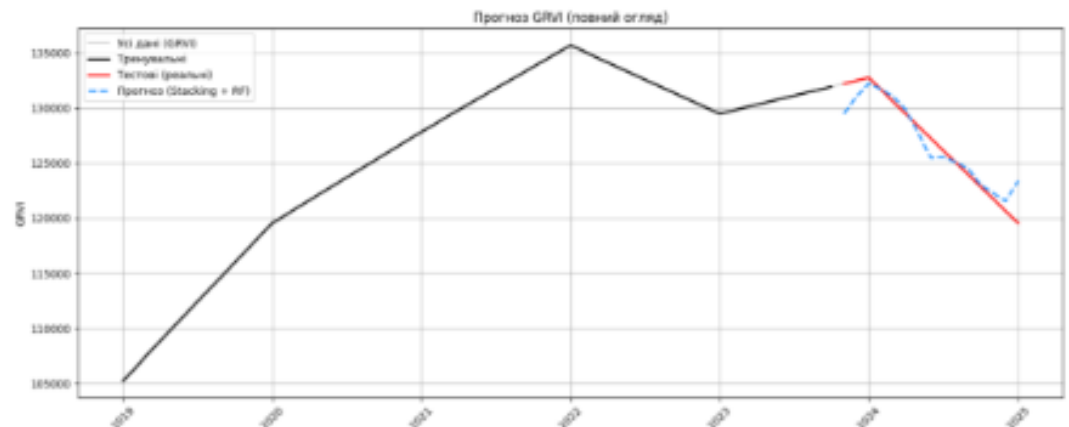


Рисунок 2 – Результат навчання Stacking моделі для ГРВІ

Інтерфейс (див. рисунок 3) модуля дозволяє користувачам вибирати тип захворювання, регіон та переглядати прогнози у вигляді інтерактивних графіків.



Рисунок 3 – Інтерфейс користувача

Тестування модуля підтвердило стабільність та зручність використання його. У майбутньому планується розширити функціональність модуля, додавши підтримку нових типів захворювань та інтеграцію з додатковими джерелами даних.

Отже, запропоновано програмний модуль для прогнозування інфекційних захворювань із використанням методів машинного навчання. Найвищу точність продемонстрував ансамблевий підхід Stacking ($RMSE = 1427.33$, $MAPE = 0.78\%$), що значно перевищує результати класичних моделей. Модуль має зручний інтерфейс, підтримує вибір захворювання та регіону, а результати подаються у вигляді інтерактивних графіків. У перспективі передбачено розширення функціоналу та інтеграцію з додатковими джерелами даних.

Список використаних джерел

1. Yang S., Santillana M., Kou S.C. Accurate estimation of influenza epidemics using Google search data via ARGO // *Proceedings of the National Academy of Sciences*. – 2015. – Vol. 112, No. 47. – P. 14473–14478.
2. Liu Y., Xu Z., Fan C., Xu L. Deep learning forecasting for influenza trend using attention-based recurrent neural networks // *IEEE Access*. – 2020. – Vol. 8. – P. 46836–46845.
3. Ribeiro M.H.D.M., da Silva R.G., Mariani V.C., dos Santos Coelho L. Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil // *Chaos, Solitons & Fractals*. – 2020. – Vol. 135. – Article ID: 109853.