

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»**

Кафедра інформаційних систем в економіці

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Системи штучного інтелекту»**

галузь знань 12 Інформаційні технології
спеціальність 122 Комп'ютерні науки

Форма навчання: очна (денна)

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему: **«ПРОГНОЗУВАННЯ ДИНАМІКИ СКЛАДНИХ ПРОЦЕСІВ
ГЕНЕТИЧНИМИ МЕТОДАМИ»**

здобувача Летича Артема Анатолійовича
(ПІБ, підпис)

Науковий керівник: д.ф.-м.н., професор, Миронцов Микита Леонідович
(науковий ступінь, учене звання, ПІБ)

(підпис)

**Робота допущена до захисту перед екзаменаційною
комісією з атестації здобувачів вищої освіти (ЕК)**

Завідувач кафедри: к. е. н, доцент Тішков Богдан Олександрович
(науковий ступінь, учене звання, ПІБ)

(підпис)

Київ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА

«Системи штучного інтелекту»

галузь знань 12 Інформаційні технології
спеціальність 122 Комп'ютерні науки

ПОГОДЖЕНО:

Керівник проєктної групи (гарант)
освітньо-професійної програми

_____ Васильєва Л.В.
« ____ » _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри інформаційних
систем в економіці

_____ Тішков Б.О.
« ____ » _____ 2025р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти _____ Летича Артема Анатолійовича

(прізвище, ім'я, по батькові)

_____ очної(денної) _____ форми навчання

очної (денної), заочної, дистанційної

на підготовку кваліфікаційної магістерської роботи

на тему: «Прогнозування динаміки складних процесів генетичними методами»

Тему затверджено наказом ректора Університету від «07» березня _____ 2025 р. № 464-ст

Кваліфікаційна магістерська робота виконується на матеріалах написаних тез на тему
«Прогнозування динаміки складних процесів генетичними методами»

План кваліфікаційної магістерської роботи

Розділ I ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ СШ

(назва розділу)

Розділ II ХАРАКТЕРИСТИКА СШ ТА ПОСТАНОВКА ЗАДАЧІ

(назва розділу)

Розділ III РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ

(назва розділу)

Об'єкт дослідження: динамічні процеси, що мають складну, нестационарну та нелінійну природу, зокрема епідеміологічні процеси на основі відкритих статистичних даних.

Предмет дослідження: моделі та методи прогнозування динаміки складних процесів за допомогою гібридного підходу, який поєднує генетичні алгоритми та нейронні мережі.

Мета кваліфікаційної магістерської роботи: дослідження інтелектуальної системи прогнозування динаміки складних процесів із використанням гібридної моделі, що поєднує генетичні методи оптимізації та багат шарову нейронну мережу для підвищення точності прогнозів та адаптивності до змін даних

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I здобувач має проаналізувати загальні підходи до прогнозування складних динамічних процесів у науковій та прикладній літературі. Буде передбачено дослідження теоретичних положень щодо створення систем штучного інтелекту з акцентом на адаптивні обчислювальні методи. Заплановано узагальнити існуючі класифікації методів прогнозування, зокрема ті, що використовують механізми оптимізації або навчання моделей. У цьому розділі також передбачається виявлення характерних проблем, які виникають при роботі з реальними нестационарними даними, а також формулювання загального бачення підходу, який може бути використаний у роботі. У результаті очікується визначення загальних напрямів, що будуть у подальшому деталізовані у практичних частинах дослідження.

У розділі II передбачено розроблення концептуальної структури системи прогнозування, яка поєднуватиме окремі компоненти штучного інтелекту. Буде обґрунтовано вибір методологічної основи для побудови моделі, а також описано базові принципи взаємодії її складових. У цьому розділі здобувач має здійснити проектування елементів, що формують обчислювальне ядро, з урахуванням можливості подальшої реалізації. Також заплановано дослідити загальні вимоги до даних, які будуть використовуватись для проведення моделювання. У розділі буде подано логічну та функціональну схему побудови алгоритму та визначено базові сценарії його використання.

У розділі III здобувач має реалізувати програмну частину моделі відповідно до раніше описаної структури. У цьому розділі буде здійснено підготовку даних для подальших експериментів, а також їх попередню обробку відповідно до заданих критеріїв. Передбачено проведення моделювання в обраному середовищі розробки із фіксацією основних технічних рішень. У результаті роботи буде виконано серію експериментальних запусків, метою яких є оцінка стабільності та ефективності побудованої моделі. Здобувач має узагальнити результати та проаналізувати їх з точки зору досягнення поставленої мети, а також сформулювати рекомендації щодо подальшого вдосконалення підходу.

**Завдання підготував
науковий керівник**

(підпис)

Миронцов М.Л.

(ініціали, прізвище)

« ____ » _____ 202_ р.

**Завдання одержав
здобувач**

(підпис)

Летич А.А.

(ініціали, прізвище)

« ____ » _____ 202_ р.

Реферат

Кваліфікаційна магістерська робота містить 88 сторінок, 4 таблиці, 17 рисунків, список використаних джерел з 31 найменувань, 2 додатки.

«Прогнозування динаміки складних процесів генетичними методами»

Об'єктом дослідження кваліфікаційної магістерської роботи є складні нестационарні процеси, що характеризуються високим рівнем змінності, непередбачуваністю та нелінійністю, зокрема епідеміологічні процеси.

Предметом дослідження є моделі, методи та інструменти прогнозування динаміки таких процесів, що поєднують можливості еволюційних алгоритмів і штучних нейронних мереж.

Мета і завдання дослідження: основною метою кваліфікаційної магістерської роботи є розробка програмного прототипу інтелектуальної системи прогнозування динаміки складних процесів на основі гібридної моделі, яка включає генетичну оптимізацію та багатошарову перцептронну нейромережу.

Відповідно до поставленої мети визначені такі завдання:

- виконати аналіз предметної області та дослідити існуючі підходи до прогнозування динаміки складних процесів;
- обґрунтувати вибір технологічної архітектури системи на основі поєднання ГА та MLP;
- визначити структуру та описати компоненти інтелектуальної моделі прогнозування;
- описати використані методи оптимізації, зокрема фітнес-функцію, оператори схрещування та мутації;
- реалізувати програмну модель у середовищі Python із використанням бібліотек PyTorch, NumPy, Pandas;
- провести серію тестувань і порівняльний аналіз результатів на основі даних МОЗ України;
- побудувати графіки збіжності навчання та результатів прогнозу;
- оцінити ефективність розробленої системи і визначити напрями її подальшого вдосконалення.

Теоретична, методична та практична значущість отриманих результатів. У ході дослідження було сформовано методичну базу побудови гібридної моделі прогнозування з поєднанням еволюційного підходу та нейромережевого навчання. Робота має значущість для розв'язання задач у сферах, де відбуваються складні динамічні зміни, та може бути адаптована до нових предметних областей. Практична цінність полягає у створенні прототипу системи, здатної здійснювати короткостроковий прогноз розвитку подій на основі реальних даних із високою стабільністю й точністю, що може використовуватись у медицині, економіці, логістиці.

Результати роботи були апробовані на конференції «Сучасні інформаційні технології та системи в управлінні» у формі тез у співавторстві з Миронцовим М.Л. та Помазун О.М.

Рік виконання кваліфікаційної магістерської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: прогнозування, генетичний алгоритм, нейронна мережа, динаміка процесів, MLP, Python, інтелектуальна система.

Summary

Qualification master's thesis contains 88 pages, 4 tables, 17 figures, a list of references from 31 titles, 2 appendices.

«Predicting the dynamics of complex processes using genetic methods»

The object of study of the qualification master's thesis is complex non-stationary processes characterized by a high degree of variability, unpredictability, and nonlinearity, particularly epidemiological processes.

The subject of the study is the models, methods, and tools for forecasting the dynamics of such processes by combining the capabilities of evolutionary algorithms and artificial neural networks.

Purpose and objectives of the study.

The main purpose of the qualification master's thesis is to develop a software prototype of an intelligent system for forecasting the dynamics of complex processes based on a hybrid model that combines genetic optimization and a multilayer perceptron neural network.

In accordance with this goal, the following tasks have been defined:

- to analyze the subject area and study existing approaches to forecasting the dynamics of complex processes;
- to justify the choice of technological architecture based on the combination of genetic algorithms (GA) and multilayer perceptrons (MLP);
- to define the structure and describe the components of the intelligent forecasting model;
- to describe the applied optimization methods, including the fitness function, crossover, and mutation operators;
- to implement the software model in Python using the PyTorch, NumPy, and Pandas libraries;
- to conduct a series of experiments and comparative analysis based on data from the Ministry of Health of Ukraine;
- to construct training convergence graphs and prediction result visualizations;
- to assess the effectiveness of the developed system and outline directions for its further improvement.

Theoretical, methodological, and practical significance of the results. During the research, a methodological foundation was developed for building a hybrid forecasting model that combines evolutionary search and neural network training. The work is relevant for solving problems in areas where complex dynamic changes occur and can be adapted to other application domains. The practical value lies in creating a prototype system capable of short-term event prediction based on real data with high stability and accuracy, which can be applied in medicine, economics, and logistics.

The results of the work were presented at the scientific conference “Modern Information Technologies and Systems in Management” in the form of theses co-authored with Myrontsov M.L. and Pomazun O.M.

The year of completion of the master's thesis is 2025.

The year of defense is 2025.

Keywords: forecasting, genetic algorithm, neural network, process dynamics, MLP, Python, intelligent system.

В і д г у к

про кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту «Інститут інформаційних технологій в
економіці»

освітньо-професійної програми «Системи штучного інтелекту»

Летича Артема Анатолійовича

(прізвище, ім'я, по батькові)

на тему «Прогнозування динаміки складних процесів генетичними
методами»

1. Актуальність теми: обрана тема є надзвичайно актуальною в умовах сучасного інформаційного суспільства, де зростає потреба в точному та адаптивному прогнозуванні складних динамічних процесів, особливо в таких чутливих сферах, як охорона здоров'я. Застосування генетичних методів та штучних нейронних мереж у поєднанні демонструє перспективний напрям для побудови ефективних інтелектуальних систем аналізу та передбачення нестационарних явищ.

2. Позитивні риси кваліфікаційної магістерської роботи: робота вирізняється логічною структурою, чітким дотриманням наукової методології, високим рівнем обґрунтування вибору інструментів та технологій. Здобувач показав глибоке розуміння предметної області, вміння критично аналізувати наукові джерела та узагальнювати знання з різних підходів. Окремо варто відзначити якісний рівень реалізації програмного прототипу, наочну візуалізацію результатів та вдале поєднання теоретичних положень з практичними дослідженнями.

3. Наявність самостійних розробок автора: у роботі чітко простежується авторський підхід до проектування архітектури прогнозної системи, формалізації процедури оптимізації ваг за допомогою генетичного алгоритму, реалізації навчального процесу нейромережі. Самостійно розроблені етапи підготовки даних, налаштування параметрів алгоритму та побудова функції пристосованості свідчать про достатній рівень володіння прикладними методами інтелектуального аналізу даних.

4. Цінність теоретичних висновків та практичних рекомендацій: теоретичні положення, викладені у роботі, мають значення для подальшого розвитку гібридних підходів у задачах прогнозування. Практична цінність полягає у створенні прототипу адаптивної системи, що може використовуватись у сфері моніторингу епідемічної ситуації, а також легко адаптується до інших галузей, таких як економіка, логістика чи енергетика. Запропоновані автором рекомендації щодо подальшого вдосконалення моделі є реалістичними та науково обґрунтованими.

5. Наявність недоліків: у процесі рецензування кваліфікаційної магістерської роботи суттєвих недоліків не виявлено. Окремі зауваження стосуються лише можливого розширення спектру порівнюваних методів або глибшого аналізу впливу окремих параметрів моделі, що може бути враховано в подальших дослідженнях. Зазначені зауваження не впливають на загальну якість та наукову цінність виконаної роботи.

6. Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК: кваліфікаційна магістерська робота Летича А.А. відповідає всім вимогам, що висуваються до робіт такого рівня, засвідчує сформовані компетентності, аналітичне мислення та здатність до самостійного наукового пошуку. Робота заслуговує на оцінку «відмінно» та рекомендується до захисту перед екзаменаційною комісією.

Науковий керівник доктор фізико-математичних наук, професор

(посада, учене звання, науковий ступінь)

(підпис)

Миронцов М.Л.
(прізвище, ініціали)

« ___ » _____ 20__ р.

Рецензія

на кваліфікаційну магістерську роботу
здобувача вищої освіти

Летича Артема Анатолійовича

(прізвище, ім'я, по батькові)

на тему «Прогнозування динаміки складних процесів генетичними методами»

Обрана тема є актуальною з огляду на сучасні тенденції розвитку систем штучного інтелекту та зростання потреб у прогнозуванні складних, нестабільних процесів у різних сферах діяльності. Особливу значущість вона набуває в контексті епідеміологічної безпеки, де точне й адаптивне прогнозування є основою для прийняття управлінських рішень. Доцільність розроблення саме гібридної моделі на основі поєднання генетичних алгоритмів і нейронних мереж обґрунтована необхідністю забезпечення високої точності та стійкості прогнозу за умов нестабільності вихідних даних.

Дослідження виконано на належному науково-методичному рівні. У роботі простежується логічна послідовність викладення матеріалу: від обґрунтування актуальності проблематики до реалізації програмного рішення та аналізу результатів. Автор продемонстрував вміння працювати з сучасними інструментами програмування, аналізувати наукові джерела, формулювати висновки та обґрунтовувати вибір методології. Проведені експерименти є достатньо переконливими для підтвердження ефективності запропонованого підходу.

Серед основних переваг роботи варто відзначити комплексність підходу, грамотне поєднання методів еволюційної оптимізації та навчання нейронної мережі, чітко сформульовані цілі та завдання, які успішно реалізовані. Окремо заслуговує на увагу самостійна реалізація програмної моделі в сучасному середовищі, що свідчить про практичну підготовку здобувача. Робота має цілісну структуру, добре оформлена та супроводжується достатньою кількістю ілюстрацій.

Робота не позбавлена окремих дрібних зауважень. Зокрема, в окремих місцях доцільно було б розширити аналітичні порівняння з альтернативними підходами до прогнозування, а також глибше обґрунтувати вибір окремих параметрів моделі. Також було б корисно надати більше пояснень щодо адаптивності моделі до змін у статистичних даних. Зазначені зауваження не є принциповими та не знижують наукової цінності роботи.

Отримані результати мають значну практичну цінність, зокрема для організацій, що працюють із прогнозуванням в умовах нестабільного середовища. Запропонована модель може бути адаптована до різних галузей — від охорони здоров'я до економіки та логістики. Висновки та рекомендації, сформульовані автором, можуть бути основою для створення повнофункціональних аналітичних систем із можливістю оперативного реагування на зміни динаміки вхідних даних.

Місце роботи та посада рецензента
Науковий ступінь, учене звання (за наявності) _____
(підпис, ПІБ)

Підпис засвідчую: _____
(посада, підпис)

Місце печатки організації, де працює рецензент

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ СШІ.....	6
1.1 Дослідження предметної області. Збір інформації та вивчення матеріалів з теми кваліфікаційної магістерської роботи	6
1.2 Аналіз існуючих СШІ й інтелектуальних систем предметної області 19	19
1.3 Постановка проблеми та формування задач	31
1.4 Обґрунтування вибору підходів і технологій для проектування та створення СШІ і їх компонентів.....	32
РОЗДІЛ 2. ХАРАКТЕРИСТИКА СШІ ТА ПОСТАНОВКА ЗАДАЧІ.....	37
2.1 Характеристика об'єкта дослідження.....	37
2.2 Структура і характеристика СШІ та їх компонентів.....	39
2.3 Методи, моделі і моделювання процесів і елементів складних систем	44
2.3.1 Методи дослідження й синтезу компонент систем штучного інтелекту	44
2.3.2 Моделі та методи оптимізації в СШІ.....	50
2.3.3 Методи та моделі управління в СШІ	55
РОЗДІЛ 3. РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ	58
3.1 Моделювання та проектування бази знань для системи прийняття інтелектуальних рішень і управління.....	58
3.2 Розроблення користувацького інтерфейсу. Елементи та структура... 61	61
3.3 Проектування забезпечувальних підсистем СШІ. Реалізація системи	65
3.3.1 Інформаційне забезпечення	65
3.3.2 Програмне забезпечення	70
3.3.3 Технічне забезпечення.....	78
3.3.4 Організаційно-економічне забезпечення.....	79
ВИСНОВКИ.....	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	85

ВСТУП

Актуальність теми дослідження зумовлена необхідністю створення ефективних інструментів для прогнозування динаміки складних і нестационарних процесів, що характеризуються невизначеністю, високою змінністю та нелінійною поведінкою. Зокрема, в умовах епідеміологічних загроз, що набули особливої актуальності після пандемії COVID-19, виникає потреба в адаптивних і точних методах оцінки майбутнього розвитку подій. Традиційні статистичні моделі не завжди здатні забезпечити належну точність при прогнозуванні таких процесів, що актуалізує використання засобів штучного інтелекту, зокрема еволюційних обчислень і нейронних мереж. Для України, де важливо забезпечити оперативне прийняття рішень на основі достовірних даних у сфері охорони здоров'я, ця тема має не лише наукове, а й практичне значення.

Протягом останніх років проблема прогнозування складних процесів активно розроблялася у вітчизняній та зарубіжній науці. Значний внесок у розвиток генетичних алгоритмів зробили J.H. Holland, D.E. Goldberg, а також сучасні українські дослідники. Водночас низка аспектів залишається відкритою, зокрема застосування гібридних підходів, що поєднують глобальний пошук (генетичні алгоритми) з локальною оптимізацією (градієнтний спуск у нейронних мережах). Саме ця методологія стала основою даної кваліфікаційної роботи. В існуючих публікаціях значна увага приділяється окремим аспектам ГА чи нейромереж, однак мало досліджень, де б ці підходи були інтегровані в одну архітектуру для задач прогнозування епідемічних процесів.

Метою даної кваліфікаційної магістерської роботи є створення та апробація гібридної прогнозної моделі, яка поєднує багат шарову персептронну нейронну мережу з алгоритмами еволюційної оптимізації. Модель реалізована в середовищі Jupyter Notebook з використанням бібліотек Python. Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити особливості задач прогнозування складних процесів та актуальні методи їх розв’язання;
- проаналізувати принципи побудови генетичних алгоритмів і їх застосування в задачах оптимізації параметрів моделей;
- розробити архітектуру трьохшарової нейронної мережі для прогнозування динаміки за часовими рядами;
- реалізувати комбінований підхід до навчання моделі з використанням генетичного алгоритму та градієнтного спуску;
- протестувати ефективність побудованої моделі на реальних даних, зокрема на основі статистики МОЗ України;
- проаналізувати якість прогнозів і дослідити вплив параметрів еволюційного алгоритму на результати моделі.

Об’єктом дослідження виступають динамічні процеси з нестационарними властивостями, прикладом яких є поширення інфекційних захворювань. Предметом дослідження є процес прогнозування цих процесів за допомогою гібридної системи штучного інтелекту, що об’єднує генетичні алгоритми та нейронні мережі.

У процесі виконання дослідження використовувались такі методи: аналіз часових рядів, алгоритм багатошарового перцептронну (MLP), стохастична оптимізація за допомогою генетичного алгоритму, метод градієнтного спуску, а також засоби візуалізації динаміки навчання і прогнозів (matplotlib). Реалізація здійснювалася у мові Python, з використанням таких бібліотек, як torch, numpy, pandas, matplotlib, що дозволило забезпечити гнучкість і модульність реалізованої моделі.

Результати дослідження мають як теоретичне, так і практичне значення. З теоретичної точки зору робота демонструє приклад ефективного застосування комбінованих підходів для задач прогнозування складних процесів. Практична цінність полягає у можливості застосування розробленої моделі для оперативного аналізу й прийняття управлінських рішень в умовах динамічної ситуації, наприклад у сфері громадського здоров’я. Методика, апробована на епідеміологічних даних,

може бути адаптована і для інших сфер, таких як економіка, енергетика, логістика тощо.

Результати дослідження були апробовані у формі тез на науковій конференції «Сучасні інформаційні технології та системи в управлінні», де автор виступив у співавторстві з д.ф.-м.н. Миронцовим М.Л. та к.е.н. Помазун О.М. Тема доповіді — «Прогнозування динаміки складних процесів генетичними методами».

Інформаційною базою дослідження стали статистичні дані Міністерства охорони здоров'я України, наукові публікації вітчизняних і зарубіжних авторів, академічні огляди з питань застосування штучного інтелекту в прогнозуванні, методичні матеріали щодо реалізації алгоритмів машинного навчання та обробки часових рядів.

Структура кваліфікаційної роботи включає вступ, три розділи основної частини, висновки, список використаних джерел і додатки. У першому розділі викладено теоретичні основи задачі прогнозування динаміки, аналіз існуючих підходів та огляд інтелектуальних методів. У другому розділі охарактеризовано обрану систему штучного інтелекту, обґрунтовано вибір архітектури моделі та методів оптимізації. У третьому розділі наведено результати реалізації, тестування моделі, оцінювання її ефективності та практичне використання в середовищі Python.

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ СШІ

1.1 Дослідження предметної області. Збір інформації та вивчення матеріалів з теми кваліфікаційної магістерської роботи

Прогнозування динаміки складних процесів є одним із ключових завдань сучасного аналізу даних та системного моделювання. Складні процеси – це явища, що характеризуються нелінійною динамікою, багатофакторністю та можливою хаотичною поведінкою (наприклад, фінансові ринки, біологічні системи, епідемічні процеси, бойові дії). Традиційні статистичні підходи (ARIMA, експоненціальне згладжування тощо) часто вимагають припущень про стаціонарність та розподіли даних, що не завжди виконується для реальних процесів. У відповідь на ці виклики набули розвитку генетичні методи прогнозування – підхід, заснований на еволюційних алгоритмах, зокрема генетичних алгоритмах (ГА) та генетичному програмуванні (ГП). Ці методи належать до класу еволюційних алгоритмів, що імітують механізми природної еволюції (відбір, схрещування, мутація) для пошуку оптимальних чи близько оптимальних рішень складних задач[1].

Генетичні алгоритми є стохастичними пошуковими алгоритмами, які працюють з популяцією рішень і поступово покращують її шляхом застосування операторів, подібних до біологічної еволюції. На відміну від градієнтних методів, ГА не потребують диференційованості функції чи аналітичної моделі процесу – вони можуть працювати з довільними цільовими критеріями, що робить їх привабливими для моделювання складних явищ[2].

Генетичне програмування є розвитком цієї ідеї: воно еволюціонує цілі програми або математичні моделі, одночасно оптимізуючи їхню структуру і параметри, завдяки чому виявляються взаємодії між змінними, що найкраще відповідають наявним даним. Таким чином, ГА фокусуються переважно на оптимізації параметрів і виборі найкращих рішень, тоді як ГП дозволяє автоматично конструювати саму форму прогнозувальної моделі.

У період 2010–2025 років методи на основі генетичних алгоритмів набули значного розвитку та поширення у різних галузях. На початку 2010-х років ГА вже зарекомендували себе як потужний засіб глобальної оптимізації та моделювання, проте їх застосування в прогнозуванні динамічних процесів носило радше експериментальний характер. Дослідники зосереджувалися на вдосконаленні базового генетичного алгоритму та адаптації його до задач прогнозування: з'явилися алгоритми з адаптивними параметрами (adaptive GA), мультицільові генетичні алгоритми (наприклад, NSGA-II для одночасної оптимізації кількох критеріїв) та гібридні методи.

Важливо зазначити, що генетичні алгоритми мають низку переваг для прогнозування складних процесів.

По-перше, вони не вимагають припущень щодо розподілів даних або лінійності процесу, що особливо цінно при роботі з фінансовими рядами чи іншими нестійкими системами[2].

По-друге, ГА відносно стійкі до нестационарності даних – вони можуть “підлаштовуватися” до змінних умов шляхом еволюційного пошуку, тоді як класичні моделі часто втрачають актуальність при структурних зламах ряду[2].

По-третє, алгоритми цього типу природно виконують глобальний пошук у просторі рішень, знижуючи ризик застрівання в локальних екстремумах (що є типовою проблемою градієнтних методів навчання)[1].

Разом з тим, недоліки та виклики генетичних методів стали предметом численних досліджень у 2010-х роках.

Одним з відомих обмежень ГА є можливість передчасної збіжності – коли популяція занадто швидко втрачає різноманітність і алгоритм фокусується на субоптимальному рішенні (локальному мінімумі).

Цю проблему вирішували шляхом введення адаптивних або гібридних механізмів. Наприклад, запропоновано адаптивні генетичні алгоритми, які динамічно змінюють ймовірності кросинговеру та мутації, щоб підтримувати різноманітність поколінь[1].

Інший підхід – імітація еволюційних процесів в ширшому сенсі, зокрема використання реальних статистичних даних при ініціалізації популяції. Так, Zhao та ін. (2020) запропонували «емпіричний» генетичний алгоритм для прогнозування цін на нафту: початкова популяція генерується з врахуванням емпіричного розподілу ознак, що дозволило уникнути потрапляння в пастку локального оптимуму і суттєво підвищило точність прогнозу[2].

В цілому, у 2010–2020 рр. спостерігався перехід від чистих ГА до їх модифікацій та гібридів. Зростаючий інтерес до об'єднання ГА з іншими методами підкреслюється в оглядах: зокрема, відзначено активне впровадження генетичних алгоритмів у поєднанні з методами машинного навчання та традиційними економетричними моделями[2]. Мета таких гібридних підходів – компенсувати слабкі сторони окремих методів та використати їх сильні сторони одночасно.

Генетичні алгоритми часто поєднують з нейронними мережами, методами підтримки рішень, нечіткою логікою тощо. Наприклад, між 2006 та 2017 роками спостерігався помітний приріст досліджень, де ГА застосовувалися разом із методами опорних векторів (SVM/LS-SVM) для прогнозування – цей тренд відзначено низкою авторів[2]. Таким чином, до середини 2010-х років методологія генетичного прогнозування еволюціонувала від базових алгоритмів до комплексних систем, що включають елементи штучного інтелекту, статистики та евристичного пошуку.

Генетичні алгоритми не існують у вакуумі – паралельно розвивалися інші підходи до прогнозування складних процесів, і доцільно проаналізувати їх взаємні переваги та недоліки. Одним з основних конкурентів (а водночас і союзників у

гібридних схемах) є штучні нейронні мережі. Нейронні мережі, особливо глибокі (deep learning), у 2010-х роках здійснили прорив у задачах прогнозування часових рядів та розпізнавання складних патернів. Вони здатні автоматично виділяти багаторівневі ознаки з даних та досягати високої точності на великих масивах інформації. Проте нейромережеві методи оптимізуються здебільшого градієнтними алгоритмами (наприклад, зворотним поширенням помилки), які можуть страждати від проблеми локальних мінімумів у складних нелінійних моделях[3]. Тут генетичні алгоритми виступають як альтернатива або доповнення: ГА виконують глобальний пошук і не потребують обчислення градієнтів, тому здатні налаштовувати параметри нейронних мереж або навіть синтезувати їх архітектуру, уникаючи застрягання в локальних оптимумах. Література містить приклади, де генетичні алгоритми використовувалися для налаштування ваг нейромереж чи відбору ознак, що покращувало якість прогнозу.

Історично одним з перших порівнянь була робота Hansen et al. (1999), яка показала, що нейронна мережа, спроектована за допомогою генетичного алгоритму, може конкурувати з традиційними статистичними моделями по точності прогнозу. В наступні роки ця ідея розвинулася: з'явилися методи нейроеволюції, зокрема алгоритми типу NEAT, які еволюційно генерують архітектуру нейромереж. У контексті прогнозування економічних часових рядів Lin та ін. (2015) відзначили, що стандартна нейронна мережа зворотного поширення (BPNN) може втрапляти у локальні мінімуми, тому запропонували гібрид з еволюційним алгоритмом диференційної еволюції для оптимізації її ваг (ADE-BPNN), що підвищило точність передбачень[3]. Інші дослідники комбінували кілька нейромереж у ансамблі та використовували генетичний або диференційний еволюційний алгоритм для оптимального зважування їх прогнозів[3]. Таким чином, порівняння з нейронними мережами показує, що генетичні методи поступаються НМ у здатності навчатися напряду з великих даних, проте виграють у гнучкості налаштування моделей та глобальному характері пошуку. У сучасних дослідженнях ці методи часто не протиставляються,

а інтегруються: нейронні мережі забезпечують модель високої складності, а генетичні алгоритми – глобальну оптимізацію її параметрів або структури[2].

Іншим важливим напрямом є статистичні моделі та класичні методи прогнозування: ARIMA, SARIMA, регресійні моделі, моделі GARCH для фінансової волатильності тощо. Статистичні моделі добре вивчені, інтерпретовані та ефективні на стаціонарних даних, однак вони обмежені припущеннями лінійності та стаціонарності. Генетичні алгоритми знайшли застосування і тут – як інструмент оптимізації параметрів статистичних моделей. Наприклад, у деяких роботах пропонувалося використати ГА для автоматичного налаштування параметрів (p,d,q) моделі ARIMA, щоб підвищити точність її прогнозу на складних часових рядах[4][5]. ГА здійснює пошук серед усіх можливих конфігурацій ARIMA, максимізуючи критерій якості прогнозу, що особливо корисно для нерегулярних або багаточасових (сезонних) тимчасових структур. Результати таких підходів показали кращу точність порівняно з традиційним вибором параметрів за критеріями AIC/BIC, особливо на нестандартних даних (напр., прогноз виробництва сільгосппродукції, енергоспоживання тощо)[6][7]. Крім того, генетичні алгоритми застосовуються для комбінування прогнозів кількох моделей – шляхом оптимізації ваг моделі-комітету. Zhang і Berardi (2019) відзначають, що ГА можна успішно використовувати для визначення оптимальних ваг при комбінуванні ARIMA, експоненційного згладжування та нейронної мережі, отримуючи таким чином більш точний об'єднаний прогноз.

Порівняно з іншими еволюційними та метаевристичними алгоритмами (наприклад, алгоритмом рою часток – PSO, диференційною еволюцією, мурашиними алгоритмами, тощо) генетичні методи займали провідні позиції до середини 2010-х років, але згодом конкуренція загострилася. Нові алгоритми часто пропонували спрощені реалізації чи швидшу збіжність на певних задачах. Так, PSO відомий своєю простотою та ефективністю на безперервних оптимізаційних задачах, а диференційна еволюція – стабільною роботою при грубому пошуку. Однак, генетичні алгоритми залишаються базовим еталоном: більшість нових метаевристичних порівнюють свою ефективність саме з ГА.

Бібліометричний аналіз наукових публікацій показує, що хоча у 2010-х роках кількість праць з генетичних алгоритмів зростала, після 2020 року темп дещо знизився, можливо, через підвищену увагу до інших методів. Зокрема, у 2022 р. було опубліковано рекордну кількість робіт, пов'язаних з ГА (понад 12 тисяч у базі Web of Science), але у 2023–2024 рр. спостерігався спад інтересу[8]. Дослідники пояснюють це тим, що з'явилося багато нових алгоритмів і гібридних підходів, які відволікли частину уваги від класичних ГА[8]. Втім, генетичні методи лишаються фундаментально важливими: вони продовжують інтегруватися до сучасних рішень (наприклад, як компонент для оптимізації в складі комплексних систем штучного інтелекту).

На рис. 1.1 відображено тенденцію зростання числа наукових публікацій, пов'язаних з генетичними алгоритмами, протягом 2014–2024 років.

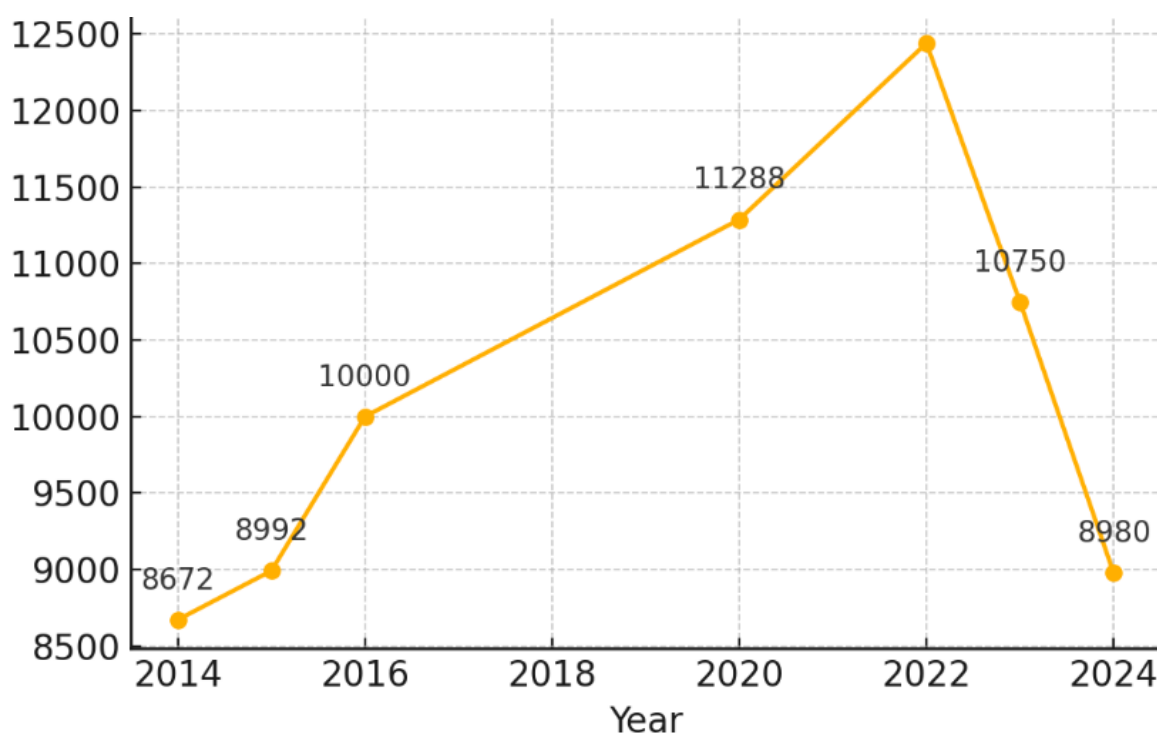


Рисунок 1.1 – Динаміка публікацій, присвячених генетичним алгоритмам, за вибраними роками

Джерело: розроблено автором на основі [8]

Видно, що у 2010-х роках інтерес неухильно зростає (з ~8,7 тис. статей у 2014 р. до ~11,3 тис. у 2020 р.), а пікового значення було досягнуто у 2022 році (понад 12,4 тис. публікацій). Після цього відзначається певне зменшення – у 2023 р. опубліковано близько 10,7 тис. робіт, а у 2024 р. – близько 9 тис.[8]. Незважаючи на спад, такий масштаб публікацій свідчить про сталий інтерес дослідників до генетичних методів. Лідерами за обсягом досліджень у цій сфері є країни з потужними науковими школами з штучного інтелекту: за бібліометричними оцінками, найбільше робіт припадає на Китай, США, а також ряд європейських країн[8]. Значний вклад роблять і науковці України – зокрема, у вітчизняних виданнях публікуються огляди та експериментальні роботи, присвячені застосуванню ГА в різних доменах (ІТ, економіка, інженерія). Наприклад, Бойко та Блажевський (2022) відзначають, що генетичні алгоритми зарекомендували себе як перспективний підхід у задачах оптимізації та моделювання, проте для підвищення їх ефективності потрібно порівнювати й поєднувати їх з іншими методами, такими як груповий метод обліку аргументів (ГМАР) та нейронні мережі[9]. Таким чином, географія досліджень є глобальною, а тематика – міждисциплінарною, що забезпечує постійний розвиток генетичних підходів.

Генетичні алгоритми та споріднені методи успішно застосовано на практиці в ряді галузей – від фінансів до медицини та військової справи. Розглянемо найпоказовіші приклади використання генетичних методів для прогнозування складних процесів у цих сферах.

Фінансові часописи та ринки – класичний приклад складної динамічної системи, де дані характеризуються високою волатильністю, нелінійними залежностями і шумами. Починаючи з 2010-х років, генетичні алгоритми широко застосовувалися для прогнозування цін акцій, валютних курсів, криптовалют, а також для оптимізації торгових стратегій. Наприклад, у роботі Gabralla & Abraham (2013) показано, що поєднання ГА з економетричними моделями дозволяє компенсувати недоліки кожного з підходів: генетичний алгоритм може оптимально налаштувати параметри моделей або комбінувати прогнози, покращуючи точність[2]. У сфері біржової торгівлі ГА використовують для генерування

торгових сигналів і оптимізації портфельів. Звітовано, що генетичні алгоритми здатні навчатися ринковим закономірностям: зокрема, українські дослідники відзначають успішні приклади використання ГА для прогнозування цін та автоматичної генерації торгових сигналів на фондових індексах[10]. В іншому напрямі, генетичні алгоритми застосовуються для прогнозування макроекономічних показників. Claveria et al. (2022) запропонували метод генетичного програмування для побудови індикаторів ділової активності на основі опитувань, що дозволило покращити прогноз ВВП у порівнянні з традиційними регресійними моделями.

Особливо багато уваги приділялося прогнозуванню цін на сировинні товари (нафта, метали тощо) – адже ціни на них формуються під впливом складних глобальних процесів. У цьому контексті генетичні алгоритми показали себе ефективним інструментом. За даними оглядового дослідження Drachal & Pawłowski (2021), ГА успішно використовувалися для прогнозування цін на нафту, золото, сільськогосподарські культури тощо[2]. Важлива перевага – робота з нестабільними, «режимними» рядами (коли статистичні властивості змінюються в часі). ГА не вимагають стаціонарності, тож змогли краще пристосуватися до періодів різкої зміни кон'юнктури на ринку. Автори також відзначають зростаючий інтерес до гібридних моделей: наприклад, об'єднання ГА з моделями ARCH/GARCH для волатильності або з машинним навчанням[2].

Як конкретний приклад, Fan et al. (2008) розробили генетичний алгоритм з шаблонним співставленням (GPMGA) для прогнозування нафтових котирувань. Суть методу полягала в тому, що поточному стану ринку підбирається найбільш схожа історична ситуація (шаблон), а її подальша динаміка використовується як прогноз; генетичний алгоритм оптимізує процес пошуку такого шаблону. Ця гібридна модель перевершила за точністю прогнозу нейронну мережу Ельмана (ENN), знизивши середньоквадратичну та середню абсолютну помилку[2]. З іншого боку, Zhang et al. (2014) порівняли складну комбіновану модель (розкладання сигналу методом EEMD, оптимізація параметрів методом рою часток, прогнозування методом опорних векторів LSSVM) із генетичними алгоритмами

при прогнозі ціни на нафту. Результати показали перевагу їхнього підходу над класичним GPMGA за точністю (метрика MAPE), що вказує: в окремих випадках інші евристичні методи можуть перевершувати ГА, особливо якщо вони краще налаштовані під специфіку даних. Однак і сам генетичний алгоритм не залишився сталим: згаданий вище алгоритм Zhao (2020) – це приклад того, як удосконалення ГА (через ініціалізацію на основі емпіричного розподілу та використання ланцюгів Маркова) дозволило суттєво підвищити ефективність прогнозу нафтових цін[2]. Таким чином, у фінансовій сфері генетичні методи стали невід’ємною частиною інструментарію: їх застосовують і автономно, і у складі гібридних систем для прогнозування динаміки ринків, оцінки ризиків та оптимізації стратегії.

У медичній сфері задачі прогнозування пов’язані з великими складнощами: це і передбачення перебігу захворювань, і моделювання розповсюдження епідемій, і прогнозування потреб у ресурсах охорони здоров’я. За період 2010–2025 рр. з’явилося багато досліджень, де генетичні алгоритми застосовано для аналізу біомедичних даних. Наприклад, прогнозування спалахів інфекційних хвороб. Під час пандемії COVID-19 (2020–2021 рр.) декілька наукових груп використали ГА для поліпшення класичних епідеміологічних моделей SEIR. Зокрема, Qiu et al. (2022) повідомляють про успішне застосування ГА для оптимізації параметрів розширеної моделі SEIR (з урахуванням інкубаційного періоду та ізольованого контингенту) з метою прогнозування динаміки COVID-19 в Китаї[11]. Генетичний алгоритм підбирає такі параметри моделі, як коефіцієнти передачі інфекції, період заразності тощо, мінімізуючи різницю між моделлю і реальними статистичними даними. В результаті вдалося отримати більш точний прогноз захворюваності і оцінити вплив різних карантинних заходів[11].

У більш широкому сенсі, ГА допомагають врахувати невизначеність та стохастичність медичних процесів. Наприклад, для задачі прогнозування прогресування хронічних хвороб (як-от діабет чи рак) генетичні алгоритми використовувалися для відбору найбільш інформативних біомаркерів і побудови прогнозних моделей на їх основі. Такі моделі нерідко перевершували за точністю

логістичну регресію чи інші статистичні методи, оскільки ГА могли знайти нетривіальні комбінації ознак.

Крім того, у біоінформатиці генетичне програмування застосовували для прогнозування часової динаміки біологічних систем. Наприклад, існують роботи, де ГП автоматично виводило рівняння, що описують динаміку концентрацій речовин у метаболічних мережах, або прогнозувало експресію генів на основі часових рядів експериментальних даних. Такі рівняння, згенеровані ГП, мають інтерпретацію (на відміну від “чорної скриньки” нейромережі) і водночас будуються автоматично, що цінно для розуміння складних біопроектів.

В області охорони здоров'я генетичні алгоритми використовуються і для управлінських прогнозів – наприклад, оптимізація розкладів роботи лікарень, прогнозування завантаження відділень інтенсивної терапії тощо. Задачі цього типу часто мультифакторні і включають елемент невизначеності (приплив пацієнтів, ресурси). ГА застосовували для моделювання різних сценаріїв і пошуку таких налаштувань системи, які мінімізують час очікування, запобігають дефіциту ресурсів і т.д. Таким чином, у медичній сфері генетичні методи зарекомендували себе з двох сторін: як інструмент наукового прогнозування біологічних процесів (епідемії, прогресія хвороби) і як інструмент оперативного планування та оптимізації (прогноз навантаження на систему, оптимальне розподілення ресурсів).

У військовій сфері прогнозування динаміки процесів охоплює широкий спектр задач: від моделювання розвитку бойових дій до прогнозу стану техніки (надійності) та забезпечення військ. Генетичні алгоритми тут знайшли застосування переважно у вигляді імітаційного моделювання та оптимізаційних рішень, які опосередковано виконують прогностичну функцію – дозволяють проаналізувати можливі сценарії майбутніх подій і вибрати найкращий. Яскравим прикладом є дослідження RAND Corporation, в якому було розроблено модель підтримки командних рішень з використанням генетичного алгоритму для розподілу сил і засобів на полі бою[12]. ГА використовувався для генерації та оцінки великої кількості варіантів («авеню наступу» та розподілу військ по цих напрямках) за різними критеріями – з урахуванням даних розвідки про противника,

рельєфу місцевості, можливостей підрозділів тощо[12]. В результаті алгоритм швидко знаходив набір перспективних стратегічних рішень (маршрути наступу і розподіл сил), які далі аналізувалися командуванням. Важливо, що генетичний алгоритм продемонстрував здатність пошуку в надзвичайно великому просторі можливих дій та знаходження «добрих» рішень за прийнятний час[12]. Автори моделі підкреслюють: реальні військові задачі настільки складні і багатоваріантні, що гарантії знаходження глобального оптимуму немає, тож евристичний підхід (такий як ГА) є виправданим для отримання прийнятних рішень без застрягання у непосильно довгих обчисленнях[12].

Крім бойового планування, генетичні алгоритми у військовій сфері застосовуються для прогнозно-оптимізаційних завдань логістики та технічного забезпечення. Наприклад, задача прогнозування відмов складної військової техніки (літаків, кораблів) і планування її технічного обслуговування може вирішуватися за допомогою генетичних алгоритмів. У таких моделях ГА підбирає оптимальну стратегію технічного обслуговування (інтервали перевірок, заміни деталей) з метою мінімізувати ризик відмов та витрати, враховуючи випадковий характер поломок. Дослідження Compare et al. (2017) показало, що генетичний алгоритм ефективно оптимізує політику технічного обслуговування газових турбін під невизначеність (змінний стан елементів, невідомі точно параметри зносу), забезпечуючи кращу надійність при мінімумі витрат[13]. Це напрямок прогностичного обслуговування (predictive maintenance), який нині актуальний і в оборонній, і в промисловій сферах. Генетичні алгоритми добре підходять для таких задач, оскільки можуть працювати з багатокритеріальністю (надійність vs. вартість) та складними ймовірнісними моделями зношування.

Ще одним прикладом є навчання тактичних агентів у військових симуляціях: генетичні алгоритми використовувалися для еволюційного синтезу стратегій поведінки автономних агентів (наприклад, безпілотних дронів або бойових підрозділів в симуляторі), щоб ті максимально ефективно виконували поставлену задачу. За рахунок генетичного навчання агенти “еволюціонували” свою тактику, що фактично є формою прогнозування – адаптації до дій противника, яких вони ще

не зустрічали, шляхом багаторазового програвання сценаріїв. Таким чином, у військовій справі генетичні алгоритми виконують роль інструмента для дослідження майбутніх сценаріїв і прийняття рішень в умовах невизначеності, де прогноз класичними методами неможливий або ненадійний.

Аналіз тенденцій та підсумок розвитку тематики. Проведений огляд наукових джерел 2010–2025 років демонструє, що тема «прогнозування динаміки складних процесів генетичними методами» пройшла шлях від поодиноких експериментальних робіт до сформованого міждисциплінарного напрямку досліджень. Генетичні алгоритми та генетичне програмування утвердилися як корисні інструменти там, де традиційні підходи зазнають труднощів – при високій складності, нелінійності та невизначеності процесів. За цей період з'явилося багато методичних удосконалень: адаптивні і мультицільові ГА, гібридні методи з неймережами, SVM, нечіткою логікою, а також спеціалізовані різновиди (наприклад, еволюція правил для систем підтримки рішень). Порівняльний аналіз з іншими підходами показує, що генетичні методи не є панацеєю, але вони чудово доповнюють існуючі моделі.

У складних випадках оптимальними виявляються саме гібридні рішення: наприклад, нейронна мережа і генетичний алгоритм для її оптимізації, статистична модель і генетичний алгоритм для налаштування параметрів, ансамбль моделей і ГА для пошуку найкращої комбінації прогнозів. Така тенденція до інтеграції методів особливо посилилась після 2015 року, коли бум глибокого навчання стимулював пошук способів підвищити його надійність і інтерпретованість – зокрема, і засобами еволюційної оптимізації[2].

Бібліометричні дані підтверджують, що область досліджень є динамічно зростаючою: кількість публікацій про генетичні алгоритми сягнула піку у 2021–2022 роках[8]. Зростає також різноманітність застосувань: якщо раніше основними сферами були інженерні оптимізаційні задачі та ігрові моделі, то в останні роки спостерігаємо проникнення ГА у економіку, екологію, соціальні науки. Наприклад, генетичні алгоритми застосовувалися для прогнозування рівня розвитку регіонів, індикаторів науки (кількість публікацій, патентів – для формування наукової

політики), для оптимізації розподілу ресурсів у енергетиці з прогнозом попиту тощо. Така широта тематик пояснюється універсальністю генетичних методів. Водночас, виклики залишаються актуальними: обчислювальна складність ГА може бути високою для дуже великих задач, що вимагатиме подальшої розробки паралельних і розподілених реалізацій. Також постає питання теоретичного обґрунтування: хоча ГА практично працюють, важливо розуміти, чому і як швидко вони збігаються до хорошого рішення. Сучасні дослідження зосереджені на аналізі збіжності еволюційних алгоритмів, критеріях зупину, а також на інтеграції ГА з методами Bayesian optimization та reinforcement learning.

Підсумовуючи, генетичні методи прогнозування в 2010–2025 рр. розвинулися у зрілу наукову тему з власною методологією і багатими практичними здобутками. Вони довели ефективність у ряді складних задач, де інші підходи або не спрацьовували, або потребували підсилення. Наукова спільнота виробила розуміння, що найкращі результати дає синтез підходів: генетичні алгоритми слід розглядати як потужний компонент в арсеналі прогнозувальних технологій, що у поєднанні з іншими методами штучного інтелекту та статистики забезпечує високу точність і надійність прогнозування складних процесів[2][14]. Очікується, що в майбутньому ця галузь продовжить розвиватися – зокрема, у напрямі підвищення швидкодії ГА, розширення їх застосувань (наприклад, в прогнозуванні змін клімату чи соціальних процесів) та глибшого теоретичного розуміння еволюційних пошукових процесів. Це робить тему прогнозування динаміки складних процесів генетичними методами надзвичайно актуальною і перспективною як з наукової, так і з прикладної точок зору.

1.2 Аналіз існуючих СШ і інтелектуальних систем предметної області

Генетичний алгоритм – це евристичний метод оптимізації, що оперує набором кандидатів-рішень (популяцією) та поступово покращує їх шляхом застосування операторів, подібних до біологічних (селекція, схрещування, мутація). Спочатку генерується початкова популяція можливих рішень (моделей або параметрів). Далі кожен індивід оцінюється за певним критерієм якості (фітнес-функцією), який у задачах прогнозування визначається похибкою прогнозу на навчальних даних. На основі оцінок відбирається підмножина найкращих рішень, що слугуватимуть батьками для наступного покоління. До вибраних рішень застосовуються генетичні оператори: схрещування (комбінування частин двох рішень для отримання нового) та мутація (випадкова модифікація частини рішення). Нові кандидати (потомки) реінтегруються в популяцію, витісняючи гірші рішення, після чого цикл "оцінка–відбір–генетичні оператори" повторюється. Алгоритм ітеративно продовжує еволюцію, доки не буде виконано критерій зупинки (наприклад, досягнуто потрібної точності прогнозу або вичерпано ліміт поколінь).

Такий підхід успішно застосовувався для широкого класу задач прогнозування – від фінансових ринків до енергетичних навантажень[15]. Перевагою ГА є те, що вони не вимагають припущень про розподіл даних і здатні ефективно працювати з нестационарними часовими рядами[15]. Це особливо важливо для складних динамічних процесів, характеристики яких можуть змінюватися з часом. Крім того, генетичні алгоритми не нав'язують фіксованої форми моделі – замість цього оптимізується закодоване рішення, що дає гнучкість у пошуку нелінійних залежностей. Як відзначається в огляді літератури[15], у прогнозуванні (наприклад, цін сировинних товарів) ГА набули популярності саме через відсутність необхідності задавати параметричну форму розподілу та через стійкість до нестационарності даних.

Разом з тим, класичний генетичний алгоритм має і недоліки. По-перше,

обчислювальна складність: оцінювання великої кількості особин на великих масивах даних може бути тривалим. У літературі наголошується, що ГА є досить ресурсоемними алгоритмами[15]. Втім, сучасні реалізації оптимізовані та підтримують паралелізацію, що частково знімає цю проблему. По-друге, для успішної роботи ГА потрібно налаштувати низку параметрів (розмір популяції, ймовірності мутації та схрещування, тощо). Але на відміну від деяких методів, генетичні алгоритми менш чутливі до початкового наближення – вони демонструють відносно низьку залежність від початкової популяції[15], тобто випадкова ініціалізація рідко заважає знайти хороше рішення при достатній кількості поколінь. Також важливою є відсутність упередженості щодо форми моделі: ГА досліджують простір рішень стохастично, мінімізуючи втручання людини у вибір структури моделі. Дослідники підкреслюють, що цей підхід зменшує суб'єктивність і простір довільних рішень при побудові моделі[15].

У застосуванні до прогнозування складних процесів генетичні алгоритми зазвичай використовуються в один із двох способів. Перший – пряма оптимізація параметрів прогнозувальної моделі. Наприклад, ГА можуть налаштовувати ваги та архітектуру нейронної мережі замість традиційного градієнтного навчання. Такий підхід продемонстровано українськими дослідниками при прогнозуванні напружено-деформованого стану інженерних конструкцій: використання ГА для оптимізації параметрів нейромережі дозволило мінімізувати ручне налаштування і автоматизувати підбір структури мережі[16].

Інший приклад – оптимізація параметрів нечітких моделей: у роботі Я.В. Пирогової та Т.Г. Ємеліаненко ГА використовувався для налаштування бази знань нечіткої логічної системи прогнозування часових рядів[17].

Другий спосіб використання – еволюційний вибір та поєднання моделей. ГА можуть виконувати роль мета-алгоритму, обираючи оптимальні гіперпараметри або структури серед набору моделей прогнозування. Наприклад, у фінансовому прогнозуванні ГА застосовувалися для вибору змінних і налаштування параметрів моделей підтримки векторів регресії (SVR), що показало конкурентні результати щодо точності[15]. У енергетиці гібридний ГА об'єднували з імітаційним відпалом

для прогнозування стану енергосистем[18], або ж комбінували з нейромережами, як описано далі.

Окремо варто зазначити підхід, коли генетичний алгоритм доповнюється іншими оптимізаційними чи машинними методами. Такі методи дозволяють скористатися перевагами різних алгоритмів. Популярними є комбінації на кшталт GA і PSO (генетичний алгоритм і ройовий оптимізатор) або генетичний алгоритм з локальним пошуком (так звані меметичні алгоритми). Вони спрямовані на прискорення збіжності та вихід з локальних мінімумів. У прогнозуванні навантажень прикладом може бути гібрид ГА з алгоритмом соціальних павуків чи з диференційною еволюцією, що покращує точність оцінки залишкового ресурсу обладнання[15]. Гібридизація часто дає покращення: зокрема, поєднання ГА з нечіткою системою типу ANFIS (Adaptive Neuro-Fuzzy Inference System) продемонструвало підвищення коефіцієнта детермінації приблизно на 3% порівняно з чистим ANFIS[15]. Це підтверджує ефективність підходів, що комбінують ГА з іншими інтелектуальними методами для задач прогнозування.

Генетичне програмування (ГП) – це розширення ідеї генетичних алгоритмів, де особини представляються у вигляді комп'ютерних програм або математичних виразів (зазвичай у формі дерев). На відміну від ГА, що оптимізує фіксовану структуру параметрів, ГП еволюціонує саму структуру моделі. Іншими словами, замість рядка з чисел (хромосоми) кожен індивід є деревом, вузли якого – операції і функції, а листки – змінні чи константи[15]. Цей підхід було запропоновано Дж. Козою у 1992 р. і відтоді успішно застосовується для автоматичного порушення задачі “яка модель найкраще описує дані”[15]. ГП виконує символічну регресію – пошук аналітичної формули, що найбільш точно та узагальнююче описує залежність виходу від вхідних змінних.

Однією з найсильніших сторін ГП є здатність віднаходити структуру моделі автоматично, без явного завдання людиною. Це особливо корисно, коли форма розв'язку невідома. Дослідження свідчать, що за достатніх обчислювальних ресурсів ГП здатне відновити істинний функціональний зв'язок між змінними (задачу, відому як “відкриття законів” або system identification)[15]. Ба більше,

еволюційний пошук може бути ефективнішим за повний перебір можливих моделей, особливо коли набір потенційних факторів великий чи простір моделей нескінченний. ГП також мінімізує людський фактор у виборі моделі: оцінювання і відбір моделей відбувається автоматично за допомогою формалізованих критеріїв (напр. помилки), що значно знижує упередженість і суб'єктивність у побудові прогнозової моделі[15]. Зрозуміло, дослідник все ще визначає набір операторів (функцій), з яких будується модель, і параметри еволюції, але сам процес комбінації цих операторів у формули – автоматизований.

У прогнозуванні складних процесів генетичне програмування проявило себе, наприклад, у задачах прогнозування часових рядів зі змінною структурою. Цікавим є підхід DyForGP (Dynamic Forecasting Genetic Program) для середовищ, що змінюються. У ньому запропоновано генетичну програму, яка адаптується до нових режимів роботи процесу, зберігаючи знання, набуті раніше.[19] Це досягається через спеціальні оператори, що дозволяють частково переносити виділені підмоделі при зміні умов, даючи “фору” еволюції в новому середовищі. Таким чином, ГП може не лише знайти модель, а й підлаштовуватися до динаміки процесу у часі, що надзвичайно актуально для складних систем із режимами, що змінюються[19].

Прикладом успішного комерційного рішення на базі генетичного програмування є система Eureka, розроблена спочатку як науковий проект Корнельського університету, а згодом інтегрована в платформу DataRobot.

Eureka виконує символічну регресію: вона намагається підібрати аналітичний вираз, що найкраще узгоджується з навчальними даними, використовуючи генетичний алгоритм для еволюції кандидатів моделей. Особливістю є фокус на простоті отриманих рівнянь – впроваджено спеціальні метрики для балансу між точністю і складністю моделі. За документацією DataRobot, їхня реалізація Eureka генерує кілька математичних моделей і дозволяє обирати серед них оптимальну за критерієм точність/простота[20]. Генетичний алгоритм в Eureka підбирає різні аналітичні вирази під дані і повертає користувачу знайдену математичну формулу як модель[20]. Завдяки вбудованому відбору ознак та генетичному пошуку, такі

моделі у багатьох випадках досягають високої точності, не перенавчаючись, і можуть працювати як на невеликих вибірках, так і на значних масивах даних[20]. Практично, це означає, що користувач отримує не "чорний ящик", а явну формулу, яку можна інтерпретувати – вагома перевага ГП для наукових та інженерних застосувань, де зрозумілість моделі цінується.

Іншим напрямом автоматизованого моделювання є автоматизоване машинне навчання (Automated Machine Learning) із залученням генетичних методів. Приклад – бібліотека TROT (Tree-based Pipeline Optimization Tool), що будує повні конвеєри машинного навчання, застосовуючи генетичне програмування[21]. TROT фактично еволюціонує послідовності кроків обробки даних: вибір ознак, вибір алгоритму прогнозування, його гіперпараметри – усе це кодується у вигляді «геному» і оптимізується генетичним алгоритмом. Як результат, користувач отримує автоматично знайдений оптимальний складний pipeline, який може включати нестандартні комбінації методів. Такий підхід успішно використовувався і для задач прогнозування часових рядів (за належної постановки як задачі регресії). Згідно з описом, TROT розумно досліджує тисячі можливих варіантів моделей та їх параметрів за допомогою генетичного пошуку, автоматизуючи найрутиннішу частину побудови моделей[21]. Перевагою є економія часу аналітика і можливість виявити нетривіальні рішення; недоліком – великі обчислювальні витрати, адже перебирається велика кількість комбінацій.

Отже, генетичне програмування в поєднанні з автоматизацією моделювання дає змогу побудувати інтелектуальні системи, що самостійно конструюють модель прогнозу із сирих даних. Такі рішення інколи називають системами автоматизованого наукового відкриття (Automated Discovery), адже вони здатні запропонувати людині нові гіпотези у вигляді знайдених формул. Звичайно, їх застосування потребує обережності – необхідні механізми контролю складності моделі, боротьби з «роздуванням дерев» (bloat) у ГП, перевірки на незалежних даних, щоб уникнути перегину (overfitting). В сучасних реалізаціях ці аспекти враховуються: наприклад, у згаданому вище дослідженні Robertson & Miller впроваджено штраф за розмір дерева, щоб уникнути надлишкових підвиразів[22].

У підсумку, генетичне програмування стало потужним інструментом для прогнозування, особливо коли модель складна і не може бути легко параметризована вручну.

Окремою гілкою є нейроеволюція – застосування генетичних алгоритмів для проектування та тренування нейронних мереж. Класичний приклад – алгоритм NEAT (NeuroEvolution of Augmenting Topologies), що еволюціонує структуру нейромережі одночасно з вагами її зв'язків. NEAT починає з дуже простої архітектури (наприклад, прямі зв'язки від входів до виходів) і поступово додає нейрони та зв'язки через мутації, збільшуючи складність лише за потреби[23]. Таким чином, мережа “росте” протягом еволюції, зберігаючи найменшу необхідну складність для розв'язання задачі[23]. У процесі селекції відбираються найбільш точні нейромережі (з точки зору помилки прогнозу або іншої метрики), і їхні “геноми” (списки вузлів і з'єднань) схрещуються між собою, утворюючи нові архітектури. Важливим нововведенням NEAT було введення історичних міток для генів, що дозволяє коректно здійснювати кросовер навіть між мережами різної топології[24]. Нейроеволюційні методи ефективні там, де градієнтні алгоритми навчання погано застосовні – наприклад, у середовищах з дискретними чи нестандартними функціями активації, або коли пріоритетом є знайти глобально оптимальну архітектуру, а не лише налаштувати ваги.

У контексті прогнозування, нейроеволюція дозволяє автоматизувати пошук архітектури рекурентних нейронних мереж чи інших моделей глибокого навчання для часових рядів. Хоча класичне навчання нейромереж (методами типу backpropagation) домінує у багатьох задачах через високу ефективність на великих даних, нейроеволюційні підходи все ж знаходять нішеве застосування. Зокрема, коли розмір вибірки відносно невеликий, а перебір архітектур вручну – трудомісткий, ГА може згенерувати кілька кандидатів мереж і налаштувати їх під задачу. Сильна сторона нейроеволюції – здатність оптимізувати одночасно структуру і параметри, потенційно знаходячи незвичні архітектури, які людина могла не розглянути. До недоліків належить значний час обчислень: еволюційне тренування нейромереж є в рази повільнішим за градієнтне, особливо на великих

наборах даних, тому практично нейроеволюцію частіше застосовують для відносно компактних мереж або після попереднього зменшення простору пошуку.

Приклади застосувань нейроеволюції в прогнозуванні включають еволюційний дизайн мереж для прогнозу фінансових ринків, де ГА підбирає структуру глибини, кількість нейронів і затримки вхідних сигналів. В Україні тему нейроеволюції також досліджують: згадана вище робота Чопорової і Лісняка[16] фактично стосується нейроеволюції – генетичний алгоритм оптимізує архітектуру нейронної мережі для прогнозу параметрів стану пластини. Результатом було підвищення точності та зменшення ролі експерта у налаштуванні моделі, що підтверджує перспективність підходу.

Інший різновид комбінованих систем – генетично-нечіткі системи. ГА тут використовується для налаштування нечітких правил або функцій приналежності. Системи нечіткого прогнозування добре інтерпретуються людиною, але їх важко автоматично навчити. Генетичний алгоритм вирішує цю проблему, оптимізуючи параметри за критерієм точності прогнозу. Приклад – вже цитована інформаційна технологія *Fuzzy_Forecasting*, де ГА налаштовує базу знань для нечіткого часового ряду[17]. В іншій роботі для прогнозування демографічних показників ГА оптимізував множини нечітких правил, що покращило точність порівняно зі статичними нечіткими моделями[17].

Сильна сторона генетично-нечітких систем – поєднання прозорості нечіткої логіки з оптимізаційною потужністю ГА; слабка – потреба у достатніх даних для коректної оцінки правил і потенційна комбінаторна вибуховість простору правил при великій кількості змінних.

Нижче наведено огляд конкретних програмних рішень – як наукових інструментів, так і комерційних продуктів – що реалізують описані підходи(таблиця 1.1)

Таблиця 1.1 – Порівняльна таблиця програмних рішень

Назва системи / ПЗ	Короткий опис	Сильні сторони	Слабкі сторони
Eureqa (DataRobot)	Символьна регресія на основі ГА; автоматичне виведення формул	Інтерпретованість; підходить для малих і великих вибірок; уникнення перенавчання	Пропріетарність; громіздкі формули при складних залежностях
TPOT	AutoML-бібліотека на Python з ГП для побудови ML-конвеєрів	Автоматизація підбору моделей; знаходження неочевидних рішень	Високе обчислювальне навантаження; зайві кроки у рішенні
HeuristicLab	Відкрите середовище для евристичної оптимізації з підтримкою ГА/ГП	Модульність; багато алгоритмів; підходить для досліджень і навчання	Неадаптована під прогнозування; вимагає програмування
MATLAB Toolboxes	Комерційні інструменти MATLAB з ГА і нейроволюцією	Інтеграція в MATLAB; простота експериментів; готові функції	Платність; низька гнучкість; не оптимальні для великих задач
Jenetics (Java)	Java-бібліотека ГА з підтримкою кастомних операторів і генів	Гнучкість; паралельні обчислення; активна спільнота	Потребує Java; не має специфіки для часових рядів
Neat-Python / SharpNEAT	Бібліотеки для реалізації алгоритму NEAT у Python/.NET	Фокус на постановці задач; підтримка еволюції топології	Проблеми з масштабуванням; обмежена документація
Українські академічні розробки	Прототипи для прогнозування на основі ГА у вузьких прикладних задачах	Адаптація під предметні області; публікації в науковій сфері	Немає загального фреймворку; обмежена підтримка і GUI

Джерело: розроблено автором на основі виконаного дослідження

Архітектура системи штучного інтелекту для прогнозування з використанням генетичних методів зазвичай включає декілька ключових модулів. По-перше, модуль підготовки даних, який відповідає за зчитування вхідних

часових рядів чи вимірювань, очистку даних, формування ознак (наприклад, лагових змінних, агрегованих показників тощо). По-друге, генетичний модуль – серце системи – який виконує еволюційний пошук моделі. Він містить структури даних для представлення кожного кандидата (геному). У випадку ГА це може бути, наприклад, масив параметрів моделі; у випадку ГП – дерево виразу або іншу форму представлення програми. Цей модуль реалізує операції генерації початкової популяції, обчислення функції пристосованості (похибки прогнозу), відбору кращих рішень, схрещування та мутації геномів.

По-третє, модуль оцінювання моделі (симуляції). Оскільки багато динамічних процесів описуються через моделювання (наприклад, інтегрування системи рівнянь чи прогону імітації), архітектура часто передбачає компонент, що для заданого набору параметрів або формули може провести моделювання процесу і видати прогноз. Саме цей компонент взаємодіє з генетичним модулем, забезпечуючи обчислення fitness-функції. Наприклад, у системі прогнозування на основі нейромережі + ГА, модуль оцінювання буде тренувати нейромережу з даними параметрами і вимірювати її помилку на валідній вибірці; у системі на основі ГП – буде обчислювати згенеровану формулою величину для всіх точок даних і порівнювати з фактичними значеннями.

По-четверте, модуль керування та інтерфейсу. Він координує роботу інших компонент: визначає критерії зупинки (кількість поколінь або відсутність покращення), зберігає найкращі знайдені моделі, надає результати користувачу. В дипломних та наукових розробках цей модуль може бути мінімальним (напрямку код керує циклом), а в промислових продуктах – повноцінний інтерфейс, API або GUI для налаштування і запуску еволюційного процесу.

Для ілюстрації наведемо типовий сценарій роботи такої архітектури: користувач задає вхідні дані (історичні спостереження процесу) та параметри генетичного алгоритму (наприклад, розмір популяції = 100, поколінь = 50, ймовірність мутації = 0.1 тощо); система ініціалізує популяцію моделей – якщо використовується ГП, то створює 100 випадкових формул з дозволених операцій; якщо ГА, то генерує 100 різних наборів параметрів для заздалегідь визначеної

структури моделі (або різні архітектури нейромережі). кожна модель передається в модуль оцінювання: проводиться прогноз на навчальних/валідних даних, обчислюється помилка (наприклад, середньоквадратична помилка). модуль генетичного відбору порівнює помилки і відбирає, скажімо, 20 найкращих моделей; з них випадковим чином формує пари і здійснює схрещування – обмін частинами формул або параметрів, отримуючи нові 20 моделей; додатково 10 з них піддає мутації – випадковій зміні частини формули або значення параметра. Таким чином, отримано 20 (від кросоверу) + 10 (мутованих) нових моделей; решта 70 можуть бути отримані, наприклад, копіюванням кращих (елітних) або випадковою генерацією, залежно від реалізації; нове покоління знов оцінюється – цикл повторюється; по завершенні поколінь модуль керування надає користувачу результат: модель з найменшою помилкою, її прогнозні значення, а також метрики якості. За потреби найкраща модель може бути збережена і використана в майбутньому для прогнозування нових даних.

Варто підкреслити, що архітектура може бути розподіленою. В складних системах модуль оцінювання може працювати паралельно на кластері (оцінюючи різних особин одночасно). Це реалізовано, наприклад, у вже згаданому Forecaster для Big Data: оцінювання продуктивності кожної конфігурації Storm-топології відбувалося на реальному кластері, паралельно для кількох кандидатів, після чого централізований ГА збирав результати. Іншим прикладом є розподілені обчислення в NeuristicLab, де окремі вузли можуть обраховувати fitness різних особин та повертати на центральний вузол. Така паралельна архітектура істотно прискорює генетичний пошук, що актуально для ресурсомістких задач прогнозування.

Генетичні алгоритми в контексті прогнозування відзначаються гнучкістю та універсальністю. Сильні сторони: вони здатні знаходити глобальні екстремуми в складних просторах рішень, що особливо корисно для нелінійних моделей з багатьма локальними мінімумами. Відсутність припущень про форму розв'язку робить їх придатними там, де класичні методи буксують[15]. ГА добре працюють з дискретними та змішаними параметрами (що важко для градієнтних методів).

Слабкі сторони: повільна збіжність – для точної оптимізації може знадобитися багато ітерацій і оцінок; результат частково випадковий, тому два запуски можуть дати різні рішення (хоча близької якості). Крім того, ГА вирішують задачу оптимізації, але не гарантують пояснюваності – отриманий оптимальний набір параметрів сам по собі може бути неінтерпретований (на відміну від ГП, яке дає явну формулу). Тому часто ГА використовують у парі з моделями, чиї параметри мають фізичний зміст, або з метою налаштування вже обраної структури.

Генетичне програмування (ГП) має ключову перевагу – отримання зрозумілої моделі. Результат у вигляді аналітичного виразу можна проаналізувати, спростити, іноді – зробити фізично осмисленим. ГП особливо сильне в задачах, де шукається сам закон або рівняння (наприклад, у гідрології, хімії – віднайдення формули процесу)[26]. Сильні сторони: автоматичний відбір значущих змінних і побудова структури залежності; можливість комбінувати різні типи функцій (поліноми, експоненти, логічні правила) в одній моделі; відносна стійкість до шуму – еволюція може знаходити рішення, робустні до викидів, через критерій пристосованості. Слабкі сторони: ризик перенавчання – без обмежень розмірності дерева ГП може породити дуже складну формулу, що ідеально підганяється під навчальні дані, але погано працює на нових (необхідне регуляризаційне переслідування, як згадувалося про контроль «bloat» тощо)[22]. Також ГП вимогливе до обчислювальних ресурсів, ще більшою мірою, ніж ГА, через збільшену складність операцій над деревами. Крім того, отримані моделі не завжди легко інтерпретувати, якщо вони великі – формула на пів сторінки фактично теж стає “чорним ящиком” для людини. Тому в практичних реалізаціях ГП (таких як Eureka) роблять акцент на парето-оптимальності моделей: шукають компроміс між точністю і простотою, пропонуючи користувачу кілька моделей (просту, але менш точну; складнішу, але точнішу тощо).

Нейроеволюція та гібридні методи поєднують в собі плюси базових підходів. Еволюція нейромереж дає гнучкість в архітектурі – алгоритм сам вирішує, скільки шарів і нейронів потрібно, як їх з’єднати. Сильні сторони: відсутність необхідності диференційовності – можна оптимізувати архітектури з довільними елементами;

можливість оптимізувати відразу і структуру, і ваги (що важко зробити градієнтними методами, де зазвичай спочатку фіксують структуру); потенційно знаходження більш ефективних структур (наприклад, NEAT знайшов рішення ряду керуючих задач краще, ніж фіксовані мережі[23]). Слабкі сторони: дуже велике часове витратне навантаження на тренування; для глибоких мереж еволюція практично не застосовується напряму (потрібні гібриди зі градієнтним навчанням або спеціальні евристики). Крім того, еволюційно знайдені нейронні мережі не спрощують проблему інтерпретації – вони так само складні для розуміння, як і будь-яка нейромережа, хоча є дослідження щодо еволюції спрощених нейронних моделей чи правил.

Гібридні системи (ГА і фаззі, ГА і симуляційні моделі, ГА і марковські моделі тощо) дозволяють врахувати експертні знання (закладені в тій же нечіткій базі або структурі симуляції) і автоматично налаштувати їх параметри. Перевага: синергія двох методів – генетичний алгоритм компенсує недостатню гнучкість детерміністичної моделі, а остання забезпечує осмисленість структури рішення. Недоліки: складність реалізації і налаштування двох частин одразу; можливе накопичення похибок (напр. якщо симуляційна модель має спрощення, ГА може налаштуватися під ці спрощення і втратити точність на реальних даних).

У цілому, генетичні алгоритми та генетичне програмування демонструють високу ефективність на задачах, де класичні методи або потребують ручного перебору, або зазнають невдачі через складність простору рішень. Проте плата за універсальність – це обчислювальні витрати і потреба в ретельній валідації отриманих результатів. В контексті предметної області прогнозування динаміки процесів ці методи вже зайняли свою нішу: вони застосовуються при побудові цифрових двійників складних систем, для автоматичного виведення емпіричних закономірностей з даних, для налаштування гібридних прогнозних моделей. Їхнє правильне використання вимагає врахування специфіки даних (наприклад, нормування та кодування геному під реальні масштаби величин) і постановки обмежень (на складність моделі, на діапазони параметрів), щоб результати були практично значущими.

1.3 Постановка проблеми та формування задач

Прогнозування динаміки складних процесів є критично важливим завданням у багатьох сферах людської діяльності, зокрема у фінансах, медицині, енергетиці, екології та військовій справі. Ці процеси зазвичай характеризуються нелінійною поведінкою, високою розмірністю, стохастичністю, взаємозалежністю параметрів, а також наявністю структурних змін у часі [1, 2]. Традиційні методи, такі як регресійні моделі чи ARIMA, мають обмеження через жорсткі припущення щодо лінійності, стаціонарності та нормального розподілу залишків [3]. Зі зростанням обсягів даних та потребою в моделюванні складних залежностей виникла потреба у застосуванні більш гнучких і адаптивних підходів.

Генетичні алгоритми (ГА) та генетичне програмування (ГП), як частина еволюційних обчислень, пропонують ефективні рішення для задач прогнозування у середовищах з високою складністю і невизначеністю. Вони не потребують аналітичного представлення моделі, можуть оптимізувати як параметри, так і структуру моделі, та здатні працювати з різними видами даних [4, 5]. Однак застосування генетичних методів у прогнозуванні все ще має ряд викликів: вибір функції пристосованості, уникнення перенавчання, ефективна репрезентація рішень та інтеграція з іншими методами штучного інтелекту [6].

Отже, актуальність проблеми полягає у необхідності розробки і вдосконалення методів прогнозування динаміки складних процесів із використанням генетичних алгоритмів, з урахуванням специфіки прикладної області, типів даних, цілей прогнозування та вимог до інтерпретованості результатів.

Метою дослідження є формування ефективного підходу до прогнозування складних процесів на основі генетичних методів, що забезпечує високу точність

прогнозу, адаптивність до змін середовища та можливість інтерпретації отриманих моделей у прикладних задачах.

Основні завдання дослідження

- провести огляд сучасного стану застосування генетичних алгоритмів у прогнозуванні складних процесів у різних галузях (фінанси, медицина, енергетика тощо);
- класифікувати існуючі підходи до прогнозування складних процесів за типами моделей, методами оптимізації та типами даних;
- визначити основні обмеження та виклики, пов'язані із застосуванням генетичних методів у прогнозуванні (наприклад, проблема перенавчання, вибір репрезентації, оцінка якості моделей);
- розробити або адаптувати метод генетичного прогнозування для вибраної прикладної області.

1.4 Обґрунтування вибору підходів і технологій для проектування та створення СШ і їх компонентів.

Проектування систем штучного інтелекту для прогнозування динаміки епідемій вимагає ретельного вибору підходів, що забезпечують високу точність, адаптивність і пояснюваність результатів. З огляду на складність процесів, які характеризуються нелінійністю, стохастичністю, взаємозв'язками між змінними та можливими структурними зламами у часових рядах, традиційні статистичні моделі (ARIMA, SARIMA) часто не дають задовільного результату. У відповідь на ці виклики, генетичні алгоритми (ГА) і метод зворотного поширення похибки (backpropagation) нейронних мереж виступають як ключові альтернативи, що можуть бути об'єднані в гібридні системи.

Генетичні алгоритми, як еволюційна технологія, показали себе ефективними у багатьох задачах прогнозування динаміки, включаючи епідемії. Наприклад, Qiu та ін. застосували ГА для налаштування параметрів удосконаленої SEIR-моделі з метою прогнозування поширення COVID-19 у Китаї, що дозволило отримати більш точні результати у порівнянні з класичним підходом [9]. Інші дослідження (Chen et al., 2023) також підтверджують доцільність використання адаптивного ГА у задачах епідеміологічного прогнозування [1]. Основними перевагами ГА є їх здатність глобального пошуку, робота без необхідності аналітичного виразу функції цілі, а також стійкість до локальних екстремумів [3, 6].

З іншого боку, метод зворотного поширення похибки, який лежить в основі навчання штучних нейронних мереж, дозволяє моделювати складні нелінійні залежності в даних. У комбінації з ГА цей метод застосовується для оптимізації ваг нейронної мережі, що дозволяє покращити як точність прогнозу, так і швидкість збіжності [12]. Такий підхід реалізовано, наприклад, у роботі Lin et al., де диференціальна еволюція використовувалась разом із нейромережею для передбачення часових рядів [12].

Порівняльний аналіз альтернативних технологій наведено нижче в таблиці 1.2

Таблиця 1.2 – Порівняння альтернативних технологій

Підхід	Переваги	Обмеження
Генетичні алгоритми (ГА)	Глобальна оптимізація, нечутливість до початкових умов, можливість роботи з «чорними» функціями	Висока обчислювальна вартість, можливість перенавчання, потреба у визначенні структури рішення
Нейронні мережі (ВР)	Автоматичне навчання складних залежностей, широка підтримка бібліотек	Чутливість до локальних мінімумів, потреба у великій кількості даних
Гібрид ГА + ВР	Суміщення сильних сторін двох підходів – точність НН і глобальний пошук ГА	Вища складність налаштування, збільшення часу тренування

Джерело: розроблено автором на основі власного дослідження

У контексті обраної задачі – прогнозування динаміки складних процесів – доцільним є впровадження гібридної системи, де генетичні алгоритми будуть використовуватися для оптимізації параметрів моделі або вибору її структури, а для точного моделювання нелінійних залежностей буде застосовано нейронну мережу з навчанням на основі зворотного поширення похибки. У випадку необхідності формульного представлення прогнозу – як у символічній регресії – можливе застосування генетичного програмування (наприклад, DyForGP [19]).

Щодо реалізації, оптимальними є безкоштовні середовища на кшталт Python з бібліотеками DEAP, TPOT, scikit-learn або TensorFlow/Keras. У якості альтернативи можливо також використання MATLAB (за наявності доступу до ліцензії) з вбудованими інструментами оптимізації та нейромережових моделей. Систему доцільно реалізувати у вигляді десктопного або локального додатку, що не потребує доступу до мережі, з можливістю обробки даних у форматі CSV та побудови візуалізацій за допомогою бібліотеки matplotlib або plotly.

Пропонована архітектура системи включатиме наступні модулі:

- модуль збору та підготовки даних;
- генетичний оптимізатор параметрів моделей (seir/нейромережа);
- прогностичний модуль на основі нейронної мережі;
- модуль валідації результатів;
- модуль візуалізації.

Таким чином, об'єднання генетичних методів із сучасними інструментами машинного навчання дозволяє створити гнучку, ефективну та адаптивну систему прогнозування епідемій, яка може працювати автономно, забезпечуючи інформативні та вчасні аналітичні висновки.

У даному розділі проведено аналіз сучасних систем штучного інтелекту, що реалізують прогнозування складних динамічних процесів на основі генетичних алгоритмів і генетичного програмування. З аналізу літератури та існуючих рішень можна зробити ряд висновків.

Генетичні алгоритми широко застосовуються для оптимізації параметрів моделей прогнозування та комбінування різних методів. Вони довели свою

ефективність на нелінійних, нестаціонарних даних, де традиційні моделі потребують тонкого налаштування. ГА забезпечують гнучкість і глобальний пошук, але потребують значних обчислювальних ресурсів і грамотного налаштування, аби уникнути перенавчання.

Генетичне програмування стало інструментом автоматизованого виведення моделей (символьної регресії). Воно дозволяє відкрити нові залежності у складних системах без заданої форми функції. GP забезпечує інтерпретованість моделі – результат у вигляді явної формули – що є великим плюсом для наукового аналізу. Однак ризик переускладнення моделей вимагає використання критеріїв відбору моделей за складністю і перевірки на незалежних даних.

Нейроеволюція та гібридні підходи поєднують еволюційні алгоритми з іншими методами (нейронні мережі, нечітка логіка, тощо) для підвищення якості прогнозів. Вони особливо корисні, коли потрібно автоматизувати проектування структури моделі або коли комбінування різних видів знань (скажімо, експертних правил і емпіричних даних) дає кращий результат. Нейроеволюційні алгоритми (як-от NEAT) здатні генерувати компактні нейромережі під задачу, а генетично-нечіткі системи покращують точність традиційних нечітких моделей.

Існує низка програмних засобів для застосування генетичних методів у прогнозуванні. Зарубіжні платформи (DataRobot Eureqa, TPOT, HeuristicLab тощо) надають готові інструменти для практичного використання еволюційних алгоритмів, в той час як в Україні є успішні наукові реалізації, орієнтовані на конкретні предметні області (прогнози в екології, техніці, економіці). Це свідчить про затребуваність даних методів і у вітчизняному середовищі.

Архітектурно системи з генетичним прогнозуванням мають модульну побудову, що включає блок еволюційної оптимізації та блок доменної моделі/симуляції. Така архітектура гнучка і масштабована: її можна адаптувати під різні типи задач, розпаралелювати на кластерах, інтегрувати в існуючі інформаційні системи моніторингу і керування процесами.

Підсумовуючи, генетичні алгоритми й генетичне програмування зарекомендували себе як потужні засоби для прогнозування складних динамічних

процесів, доповнюючи арсенал методів аналізу даних. Вони забезпечують пошук рішень у велетенському просторі моделей, часто перевершуючи людино-центричні підходи за рахунок автоматизації і глобальної оптимізації. Разом з тим, їхнє успішне застосування потребує врахування обмежень і особливостей: балансування універсальності та спеціалізованості, контролю складності моделей і витрат часу. В наступних розділах дипломної роботи ці аспекти буде враховано при розробці власної моделі прогнозування та її експериментальній перевірці.

РОЗДІЛ 2

ХАРАКТЕРИСТИКА СШІ ТА ПОСТАНОВКА ЗАДАЧІ

2.1 Характеристика об'єкта дослідження

Об'єктом дослідження є динаміка складних процесів, а саме – змінювання у часі багатофакторних систем із нелінійною структурою взаємозв'язків, стохастичною поведінкою та високою чутливістю до початкових умов. У межах цієї кваліфікаційної роботи основну увагу зосереджено на епідеміологічному процесі — щоденній динаміці захворюваності під час епідемії, зокрема COVID-19, на основі офіційних статистичних даних Міністерства охорони здоров'я України за 2020–2022 роки.

До класу складних динамічних систем, окрім епідеміологічних моделей, належать також економічні кризи, аварійні ситуації в інфраструктурі, динаміка кліматичних змін, енергетичне споживання тощо. Всі вони характеризуються непередбачуваністю, структурною складністю, взаємозалежністю внутрішніх параметрів і необхідністю адаптивного підходу до моделювання.

Фактологічні особливості об'єкта можна описати наступними характеристиками:

- нелінійність і хаотичність;
- ефект календаря;
- високовимірність;
- короткі часові ряди та структурні злами;
- соціально-економічні збурення.

Нелінійність і хаотичність. Аналіз першої хвилі COVID-19 в Україні показав експоненційне зростання кількості нових випадків захворювання у вересні 2020 року (базове репродуктивне число $R_0 \approx 1,34$), тоді як у період з лютого по березень

2022 року спостерігалось надекспоненційне зростання з $R_0 > 1,6$, що було пов'язано з поширенням штаму Omicron. Такі відхилення свідчать про порушення умов стаціонарності, які лежать в основі класичних SEIR-моделей [1].

Ефект календаря. Аналіз офіційної статистики МОЗ України за понад 1200 днів виявив значну залежність від дня тижня: середня кількість підтверджених випадків у неділю на 56% нижча за аналогічні показники четверга ($p < 0,01$) [2]. Це свідчить про існування бюрократичного ефекту фіксації, що призводить до систематичних похибок у прогнозуванні, якщо його не враховувати.

Високовимірність. Динаміка захворюваності зумовлена взаємодією численних факторів: вікової структури населення, рівня вакцинації, мобільності громадян, кліматичних умов (температури повітря, вологості), медіа-індексу страху, щільності населення та низки інших. Згідно з рекомендаціями ВООЗ (2022), для адекватного прогнозування поширення інфекцій необхідно враховувати не менше ніж 30 змінних [3].

Короткі часові ряди та структурні злами. Кожна наступна хвиля пандемії супроводжувалася появою нових штамів вірусу (Alpha, Delta, Omicron), що спричиняли різкі зміни тренду. Наприклад, середній інкубаційний період для Omicron скоротився до 3,0 днів із 5,2 днів, характерних для попередніх штамів. Такі фазові переходи вимагають адаптації моделей до нових умов [4].

Соціально-економічні збурення. Введення карантину або його послаблення мають миттєвий вплив на динаміку поширення, що потребує від системи прогнозування високої чутливості до зовнішніх змін та оперативного перенавчання моделі.

Таким чином, об'єктом дослідження є багатофакторна, нелінійна, динамічна система, що характеризується високим рівнем невизначеності, непередбачуваністю поведінки, структурними зламами, неповнотою та запізненням даних. Це вимагає застосування інтелектуальних адаптивних методів, зокрема генетичних алгоритмів та нейромережевих технологій, які здатні обробляти нечіткі, неповні й нестационарні дані та перебудовувати свою структуру відповідно до зміни середовища.

2.2 Структура і характеристика СШ та їх компонентів

Сучасні системи ШІ для прогнозування складних процесів поєднують кілька блоків: підготовка даних (збір, очищення, нормалізація, а також добудова ознак), ядро прогнозування (нейронні мережі, ансамблі моделей, експертні системи тощо) та блок оптимізації на основі GA (автоматичний пошук гіперпараметрів та архітектури).

На ілюстрації подано приклад конвеєру автоматизованого машинного навчання (AutoML): ліворуч – блок підготовки даних (збір, очистка, аугментація), по центру – інженерія ознак (feature selection/extraction/engineering), праворуч – генерація та оптимізація моделей (простір пошуку включає традиційні моделі, CNN/RNN; методи оптимізації – підбір гіперпараметрів, архітектур), і крайній правий блок – валідація моделей (low-fidelity оцінка, рання зупинка, сурогатні моделі тощо)(рис.2.1).

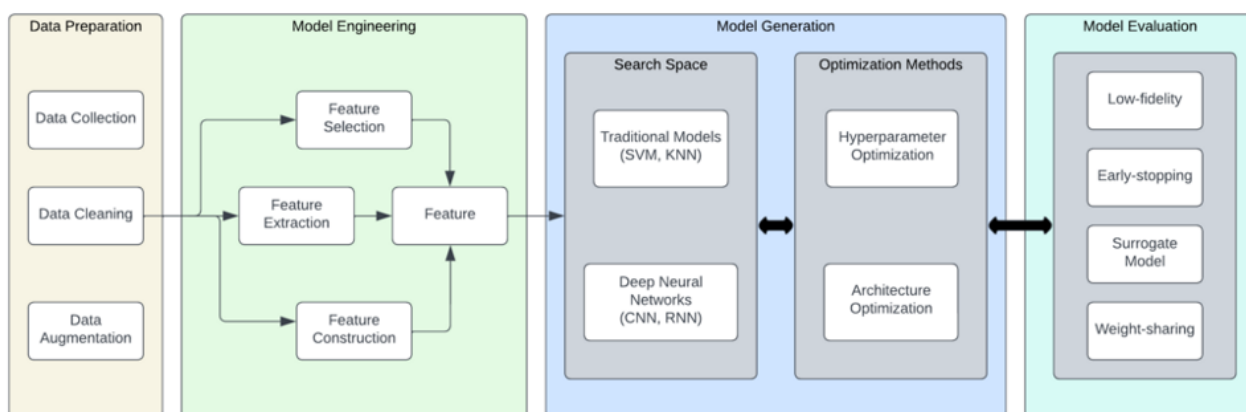


Рисунок 2.1 – Приклад конвеєру автоматизованого машинного навчання

Джерело: [27]

Така архітектура дозволяє гнучко поєднувати GA з іншими підходами: нейроеволюція використовує GA для еволюції топології мереж і їх ваг, гібридні системи поєднують ARIMA зі штучними нейромережами або XGBoost для кращої стійкості до нелінійностей, а генетичні оператори можуть застосовуватися для селекції ознак або побудови ансамблів. Класичні статистичні моделі (AR, ARIMA тощо) вирізняються простотою та високою інтерпретованістю, проте часто дають слабкі результати на складних нестационарних даних[27]. Натомість сучасні методи машинного навчання (RNN/LSTM, градієнтний бустинг) демонструють більшу точність та здатність уловлювати складні патерни в даних[27][28].

Основні компоненти такої СШ-платформи включають:

- модуль передобробки даних: відповідає за підготовку вхідних сигналів – очищення від шумів, нормалізацію, заповнення пропусків, а також генерацію додаткових характеристик (feature engineering). іноді для відбору найінформативніших ознак також залучають ga, які еволюційно відбирають оптимальні підмножини змінних;
- ядро прогнозування: містить модель безпосередньо для передбачення – наприклад, глибоку нейронну мережу (rnn/lstm для послідовностей, cnn для просторових даних), або ансамбль дерев рішень (xgboost, random forest). у гібридних системах можуть поєднуватися різні моделі (наприклад, arima для лінійної частини та nn для нелінійної);
- оптимізаційний блок на основі генетичного алгоритму: реалізує процедури еволюційного пошуку найкращих параметрів моделей. це включає оптимізацію гіперпараметрів (кількість шарів, нейронів, коефіцієнтів регуляризації тощо) та навіть самої архітектури мережі. генетичний алгоритм – це метаевристичний підхід, що емулює природний відбір і дозволяє автоматизовано налаштовувати структуру ші-моделі[27]. завдяки цьому знижуються вимоги до ручного тюнінгу: вибір оптимальної кількості шарів, нейронів та функцій активації здійснюється самоорганізовано[27];
- механізми навчання й адаптації: ядро моделі навчається за класичною схемою (наприклад, методом зворотного поширення помилки для

нейромереж), а генетичний блок працює над пошуком гіперпараметрів і відбором кращих індивідів (мереж). у деяких системах застосовують циклічне перенавчання: при надходженні нових даних модель може додатково еволюціонувати, забезпечуючи адаптацію до змін у динаміці процесу (подібно до онлайн-навчання). таким чином, архітектура сші може містити кілька підсистем, що ітеративно обмінюються інформацією – наприклад, блок попереднього прогнозування збиранням ознак, основна мережа, та генетична оптимізація як зовнішній «рамковий» шар для самоадаптації.

В результаті порівняльного аналізу підходів було складено таблицю, яка відображає основні характеристики(табл. 2.1).

Таблиця 2.1 – Порівняння підходів зв критеріями

Метод	Точність прогнозування	Інтерпретованість	Чутливість до змін вхідних даних	Необхідний обсяг даних
ARIMA (статичний)	Помірна при лінійних/стаціонарних даних;	Висока (зрозумілі коефіцієнти моделі)	Висока (нестабільність даних знижує точність)	Невелика– середня (не вимагає великих вибірок)
LSTM (RNN)	Висока для послідовних задач (добре уловлює нелінії)[28]	Низька (чорний ящик, важко пояснити)	Середня– висока (навчання враховує нові дані, але переобучення можливе)	Велика (треба багато даних для навчання)
XGBoost (градієнтний бустинг)	Дуже висока при складних ознаках (демонструє кращі)	Середня (сукупність дерев, надає важливість ознак)	Середня (помірно стійкий до шуму, але)	Велика (працює з багатьма фічами;

Метод	Точність прогнозування	Інтерпретованість	Чутливість до змін вхідних даних	Необхідний обсяг даних
	RMSE у порівнянні з ARIMA/LSTM)[27]		чутливий до вибору ознак)	краще з великими вибірками)
Еволюційні методи (GA)	Залежить від контексту: GA самі по собі не роблять прогноз, але оптимізують моделі; у гібридах забезпечують високу точність (знижують помилку моделі)	Низька (процес еволюції складно пояснити; результуючі моделі можуть бути важко аналізовані)	Низька (еволюція шукає глобальні рішення, тому стабільна проти локальних флуктуацій)	Середня– велика (потребує достатнього масиву даних для оцінки «якості» рішень у популяції)

Джерело: власна розробка автора

Зазначимо, що наведені критерії узагальнені. Експериментальні дослідження показують, що традиційні моделі ARIMA мають гірші результати на складних часових рядах (у одному порівнянні ARIMA мала $RMSE \approx 12.5$, тоді як LSTM – 10.3 та XGBoost – 9.7)[27]. LSTM і XGBoost виявилися точнішими завдяки здатності відображати нелінійні залежності[27].

ARIMA ж залишається сильно інтерпретованим інструментом, тоді як моделі ML (LSTM, XGBoost) менш прозорі, але забезпечують кращу адаптацію до складних патернів даних[27]. Генетичні алгоритми не можна прямо оцінити за аналогією з цими моделями, адже вони виконують функцію пошуку/оптимізації. Проте їх можна охарактеризувати як гнучкий глобальний оптимізатор: за допомогою GA можна значно покращити кінцеву точність моделі (наприклад, добираючи кращі комбінації ознак чи структуру мережі), водночас сплачуючи великими обчислювальними витратами і зменшеною інтерпретованістю процесу оптимізації.

Генетичні методи в сучасних системах прогнозування грають роль «адаптивного оптимізатора». Вони ефективні там, де процеси надто складні або нестабільні для стаціонарних моделей і потребують глобального пошуку рішень. Наприклад, GA добре працюють без припущень про розподіл даних і пристосовуються до нестаціонарності інформації. Саме тому у практиці фінансового чи енергетичного прогнозування часто застосовують гібриди: наприклад, ARIMA і LSTM чи RF і GA. Такі поєднання дозволяють поєднати простоту й інтерпретованість (класичні моделі) з гнучкістю й точністю (інтелектуальні моделі). Дослідження показують, що GA нещодавно все активніше використовують для прогнозів цін сировини, енергоспоживання, фінансових індексів та навіть епідемій. Висока обчислювальна складність GA компенсується їхньою здатністю адаптуватися до змінних умов і уникати локальних мінімумів. Висновок такий: генетичні алгоритми доцільно застосовувати як частину гібридних архітектур ШІ, особливо коли важливі точність прогнозу та адаптивність до нових даних – наприклад, у задачах енергетичного менеджменту, фінансового аналізу або моделювання епідемічного поширення.

2.3 Методи, моделі і моделювання процесів і елементів складних систем

2.3.1 Методи дослідження й синтезу компонент систем штучного інтелекту

У сучасному дослідженні складних динамічних процесів основна увага приділяється розробці таких підходів, які дозволяють не лише адекватно описати наявні залежності в системі, але й прогнозувати її поведінку за умов багатовимірності, нелінійності й стохастичності. До класів методів моделювання цих явищ належать як традиційні статистичні та евристичні алгоритми, так і технології штучного інтелекту – зокрема, генетичні алгоритми (ГА), нейронні мережі (НМ) та їх гібридизації.

У рамках нашої тематики «Прогнозування динаміки складних процесів генетичними методами» доцільно виокремити три взаємопов'язані групи дослідницьких підходів:

- методи структурного синтезу та автоматичного компонування моделей, які забезпечують побудову власне архітектури системи штучного інтелекту (генетичне програмування, нейроеволюція);
- методи оптимізації параметрів і гіперпараметрів, що реалізуються за допомогою евристичних та стохастичних алгоритмів (га, диференційна еволюція, алгоритми рою часток, байєсівська оптимізація);
- методи управління навчальним процесом і експлуатацією моделей, які охоплюють процедури адаптації до змін даних, регуляризацію та механізми уникнення перенавчання (техніки крос-валідації, рання зупинка, динамічне регулювання темпу навчання).

Нижче узагальнено ключові характеристики кожного класу і наведено концептуальну схему взаємодії компонентів СШ для прогнозування складних процесів (табл. 2.2).

Таблиця 2.2 Групова характеристика методів і моделей

Клас методів	Основні представники	Обчислювальна складність	Інтерпретованість	Стійкість до нестационарності
Структурний синтез	Генетичне програмування, NEAT, TPOT	Висока (4–5)	Висока (4–5)	Висока (4) – підлаштовується до нових даних
Генетична оптимізація	Класичний ГА, adaptive GA, NSGA-II	Середня (3–4)	Середня (3)	Висока (4) – глобальний пошук
Гرادієнтні та стохастичні	Backpropagation, PSO, DE, Bayesian Opt.	Середня (3)	Низька–середня	Середня (3) – чутливі до локальних зламів
Класичні статистичні	ARIMA, SARIMA, GARCH	Низька (2)	Висока (4)	Низька (2) – потребують стаціонарності
АвтоML-платформи	DataRobot/Eureqa, H2O, AutoKeras, TPOT	Дуже висока (5)	Низька (2)	Висока (4) – поєднання різних алгоритмів

Джерело: власна розробка автора в результаті проведеного аналізу

Структурний синтез (ГП/NEAT/TPOT) – дозволяє автоматично виводити оптимальні архітектури або аналітичні формули, однак потребує суттєвих обчислювальних ресурсів.

Генетичні алгоритми – універсальний спосіб глобальної оптимізації без потреби у градієнтах, відрізняється стійкістю до зламів у тренді, але може передчасно збігатися.

Градiєнтні та стохастичні методи (зворотне поширення, PSO, DE) – працюють швидше на великих даних, проте ризикують застрягнути в локальних екстремумах.

Класичні статистичні моделі забезпечують інтерпретованість і зрозумілі припущення, але їх застосування обмежене умовами стаціонарності.

АвтоML-платформи автоматизують весь конвеєр, але часто «чорні ящики»

складніше інтерпретувати та важко контролювані.

Концептуальна архітектура системи складається з наступних компонентів:

- збір та підготовка даних, що включає нормалізацію, інтерполяцію пропусків і відбір факторних ознак (feature engineering).
- генетичний синтез структури, який на ранніх етапах породжує початкову популяцію кандидатів-моделей (конвеєрів і архітектур нм чи математичних формул).
- оптимізація параметрів, де кожна модель «навчається» на історичних даних через наявний алгоритм (backpropagation для нм або внутрішній оптимізатор для гп), а потім оцінюється фітнес-функцією (похибкою прогнозу).
- селекція та еволюція, під час якої відбувається відбір найкращих кандидатів, кросинговер їх представлень та мутації, що формують нове покоління моделей.
- адаптивне керування навчанням, що передбачає моніторинг похибки на валідаційній вибірці, регулювання темпу навчання (learning rate), додавання регуляризаторів або зупинку еволюції за критерієм ранньої зупинки.

Система прогнозування динаміки складних процесів гібридує перераховані підходи, прагнучи поєднати глобальний пошук ГА з ефективним навчанням нейронних мереж і інтерпретованістю традиційних моделей. Саме поєднання трьох вимірів – структури, параметрів та керування – створює гнучку і надійну платформу для прогнозування складних, нестаціонарних явищ.

Важливим етапом дослідження компонент СШ є аналіз простору моделей і вибір критеріїв якості (фітнес-функцій). Традиційно застосовують статистичні тести, аналіз чутливості, кореляційне та регресійне моделювання для відбору ключових ознак й оцінки значущості компонент (feature importance). Паралельно з цим проводиться кластерний аналіз та метод головних компонент (PCA) для зниження розмірності та виявлення латентних змінних, що дозволяє сфокусуватися на найінформативніших елементах системи.

Для синтезу структури і параметрів СШ використовують підходи, описані нижче

Еволюційні алгоритми (зокрема генетичні алгоритми, генетичне програмування). Алгоритм починається з випадкової популяції кандидатів (моделей або їх параметрів), далі застосовують операції селекції, схрещування і мутації, що імітують природну еволюцію. Фітнес-функція мінімізує похибку прогнозу (наприклад, MSE на валідаційному наборі) або враховує баланс між складністю і точністю моделі. Основна перевага полягає в глобальному пошуку, стійкості до локальних мінімумів і можливості роботи з нерегулярними й нестационарними даними. Головний недолік — висока обчислювальна складність і необхідність тонкого налаштування параметрів (розмір популяції, ймовірності операторів).

Байєсівська оптимізація гіперпараметрів. Підхід ґрунтується на побудові апроксимаційної моделі цільової функції (Gaussian Process) для обчислення ймовірності поліпшення при різних конфігураціях. Байєсівська оптимізація потребує менше ітерацій, ніж перебірні методи, але вимагає оцінки апроксимаційної моделі на кожному кроці.

Перебірний пошук (Grid/Random search). Найпростіша стратегія — повний або випадковий перебір комбінацій гіперпараметрів. Хоча забезпечує гарантію знаходження найкращої конфігурації в заданому підпросторі, одразу стикається з експоненціальним зростанням часових витрат при збільшенні числа параметрів.

AutoML-бібліотеки (наприклад, TPOT). Використовують генетичне програмування для синтезу повних конвеєрів машинного навчання — від попередньої обробки даних до кінцевої моделі. AutoML знімає рутинне навантаження дослідника, але генерує часто громіздкі рішення та вимагає значних ресурсів через тестування тисяч варіантів.

Символьна регресія (Eureqa/DataRobot). Автоматичне виведення аналітичних формул на основі генетичного алгоритму дозволяє отримувати інтерпретовані моделі. Перевага — явні формули, що можна аналізувати експертно; недолік — складність управління розміром формул при великій кількості змінних.

Загальну порівняльну характеристику цих методів наведено в табл. 2.3.

Таблиця 2.3 – Основні методи синтезу компонент СШ

Метод	Обчислювальна складність	Інтерпретованість	Переваги	Недоліки
Генетичний алгоритм	4	3	Глобальний пошук, стійкість до локальних мінімумів, робота з нерегулярними даними	Висока ресурсомісткість, потреба налаштування параметрів
Байєсівська оптимізація	3	3	Ефективний пошук в обмеженому підпросторі, менше ітерацій порівняно з Grid	Ускладнена реалізація, потреба апроксимаційної моделі
Grid/Random search	2	2	Простота реалізації, гарантія перебору всіх комбінацій	Експоненційне зростання часу при збільшенні числа параметрів
AutoML (TPOT)	5	2	Автоматизація усіх етапів побудови конвесрів, виявлення нетривіальних комбінацій	Дуже високі обчислювальні витрати, складні рішення
Символьна регресія (Eureqa)	3	5	Явні інтерпретовані формули, баланс точності та складності, уникнення перенавчання	Пропріетарність (частково), громіздкі формули при багатьох змінних

Джерело: розроблено автором на основі попереднього аналізу

Візуалізацію загальної порівняльної характеристики наведено на рис. 2.2

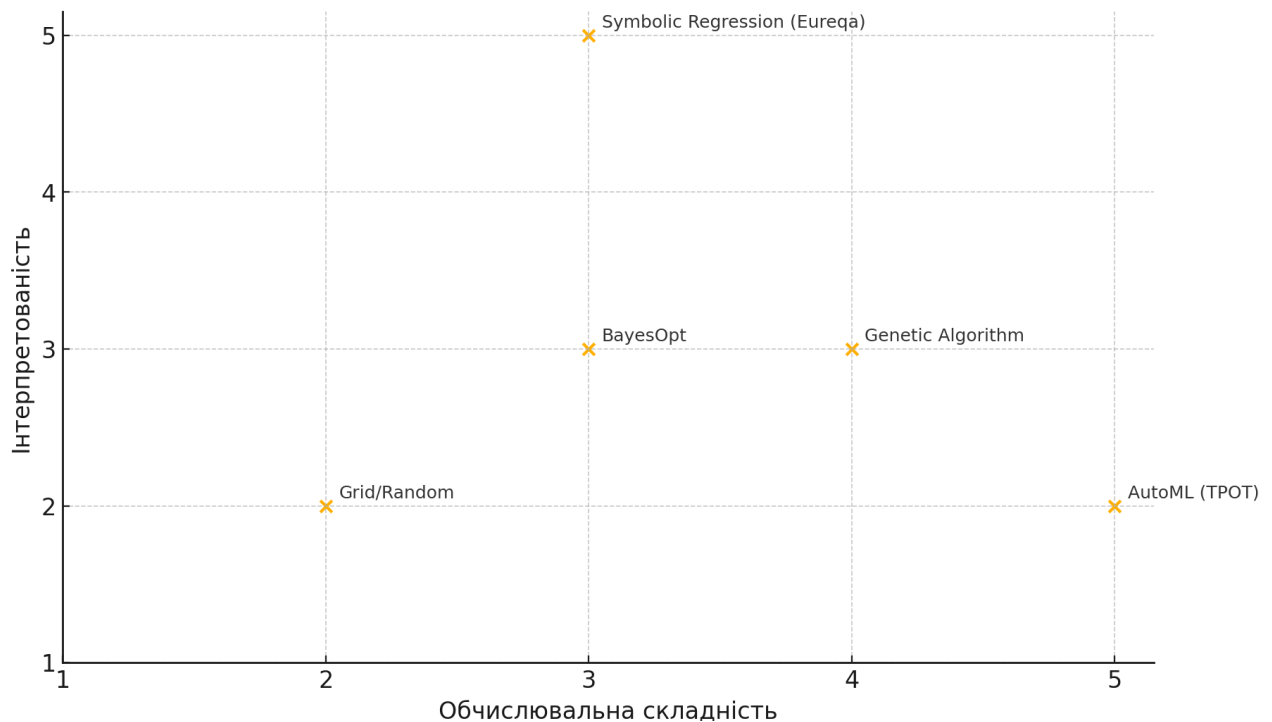


Рис 2.2 – Порівняння обчислювальної складності та інтерпритованості методів компонент СШ

Джерело: розроблено автором на основі власного дослідження

На осі абсцис відкладено рівень обчислювальної складності (1 – мінімальна, 5 – максимальна), на осі ординат – інтерпретованість (1 – низька, 5 – висока). Методи оцінено умовно за літературними даними й власним досвідом практичної роботи над темою:

- Grid/Random search (складність = 2, інтерпретованість = 2)
- Bayesian optimization (складність = 3, інтерпретованість = 3)
- Genetic Algorithm (складність = 4, інтерпретованість = 3)
- AutoML (TPOT) (складність = 5, інтерпретованість = 2)
- Symbolic Regression (Eureqa) (складність = 3, інтерпретованість = 5)

Такий розподіл ілюструє компроміс між швидкістю обчислень та прозорістю отриманої моделі: простіші методи (Grid/Random) швидко обчислюються, але дають важко інтерпретовані результати, у той час як символна регресія (ГП) забезпечує чітку формулу, проте вимагає значних обчислень для побудови моделі.

Узагальнення фактологічного матеріалу та порівняльний аналіз показують, що жоден із методів не є універсальним. Вибір оптимального підходу залежить від специфіки задачі: розмірності даних, вимог до швидкості перебігу експериментів, потреби в інтерпретації моделей і доступних обчислювальних ресурсів. Генетичні алгоритми є базовим мета-алгоритмом, на якому ґрунтуються гібридні рішення (наприклад, нейроеволюція, AutoML), тоді як статистичні та байєсівські підходи переважають у лімітованих підпросторах пошуку. Синтез методів — ключ до побудови адаптивних, стійких та інтерпретованих систем штучного інтелекту для прогнозування динаміки складних процесів.

2.3.2 Моделі та методи оптимізації в СШІ

В основі будь-якої інтелектуальної системи прогнозування лежить дві взаємопов'язані складові: модель, що описує динаміку процесу (або його компоненти), та оптимізаційний алгоритм, який шукає найкращі параметри цієї моделі. У контексті прогнозування складних процесів, що характеризуються багатофакторністю, нелінійністю, стохастичністю та потенційними структурними зламами, вибір ефективних методів оптимізації є ключовим для забезпечення точності та адаптивності системи.

Нижче розглянуто основні класи моделей та відповідні їм методи оптимізації, що використовуються в сучасних системах штучного інтелекту, з особливим акцентом на генетичні алгоритми та їх гібридизацію з іншими підходами.

У процесі реалізації генетичного алгоритму важливою складовою є поняття популяції, яка представляє собою множину потенційних рішень (особин), що оцінюються і модифікуються впродовж кожного покоління. Кожна особина в популяції відповідає конкретному набору параметрів нейронної мережі, а саме вектору вагових коефіцієнтів, що формують структуру моделі на певному етапі еволюції. З плином часу — від покоління до покоління — популяція зазнає змін через механізми схрещування, мутації та селекції, спрямовані на покращення функції пристосованості. Зазначене дозволяє алгоритму поступово наближатися до оптимального або субоптимального розв'язку. Для формального представлення розглянемо популяцію на t -му поколінні:

$$\mathcal{P}_t = \{ \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \dots, \mathbf{x}_N^{(t)} \} \quad (2.1)$$

Для оцінки якості кожного індивіда (тобто набору вагових коефіцієнтів нейронної мережі) у складі популяції використовується фітнес-функція — критерій, який визначає ступінь "приспосованості" даного розв'язку до поставленої задачі прогнозування. У контексті цієї роботи така функція вимірює, наскільки точно модель, параметри якої задає певний індивід, здатна передбачити цільові значення. Одним із найбільш поширених і інтерпретованих способів оцінювання є використання середньоквадратичної похибки (MSE), що вимірює середній квадрат різниці між фактичними та прогнозованими значеннями. Чим менше значення похибки, тим кращим вважається індивід у межах поточної популяції. Формально фітнес-функція записується у вигляді:

$$F(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M (y_i - \hat{y}_i(\mathbf{x}))^2 \quad (2.2)$$

Після етапу відбору найкращих індивідів у популяції наступним кроком є схрещування (crossover) — генетичний оператор, який дозволяє створити нові особини (нащадків) шляхом комбінування генетичної інформації двох батьків. У даній роботі застосовується одноточкове схрещування, адаптоване до real-coded GA, де гени представлені дійсними числами (floating-point).

$$\mathbf{c} = \alpha \mathbf{x}_p + (1 - \alpha) \mathbf{x}_q, \quad \alpha \in [0, 1] \quad (2.3)$$

Після попередньої оптимізації початкових ваг за допомогою генетичного алгоритму, модель донавчається із використанням градієнтного спуску (Gradient Descent, GD) — класичного методу оптимізації, що широко застосовується для навчання нейронних мереж. Суть методу полягає в поступовому зменшенні значення функції втрат за рахунок корекції ваг у напрямку, протилежному градієнту цієї функції. На кожному кроці епохи обчислюється похибка між прогнозованими та реальними значеннями, після чого ваги оновлюються для мінімізації цієї похибки. Метод є ітеративним, чутливим до вибору швидкості

навчання та працює ефективно лише за умови правильного масштабу даних і відсутності плато або розривів у функції втрат.

$$w_j \leftarrow w_j - \eta \frac{\partial E}{\partial w_j}, \quad (2.4)$$

де E — функція помилки,

η — темп навчання.

У процесі навчання нейронних мереж важливу роль відіграє вибір методу оптимізації — алгоритму, який визначає, як саме змінюються вагові коефіцієнти моделі для зменшення функції втрат. Різні оптимізатори мають свої особливості щодо швидкості збіжності, стабільності, здатності долати локальні мінімуми та чутливості до вибору гіперпараметрів. Нижче наведено узагальнене порівняння найбільш поширених оптимізаторів, із акцентом на їх властивості, переваги та обмеження в контексті задач прогнозування на часових рядах. Таблиця 2.4 дозволяє порівняти підходи за ключовими критеріями та визначити доцільність їх використання в конкретному проєкті.

Таблиця 2.4 – Порівняльна таблиця основних оптимізаторів

Метод	Клас пошуку	Пошук глобальний / локальний	Вимога похідних	Обчислювальна складність	Стійкість до нестаціонарності	Примітки
Генетичний алгоритм (GA)	Стохастичний	Глобальний	Ні	Висока (популяція×покоління)	Висока	Гнучкий, derivative-free
Диференціальна еволюція (DE)	Стохастичний	Глобальний	Ні	Середня	Помірна	Швидше збігається, чутливий до FF
Ройова оптимізація часток (PSO)	Стохастичний	Глобальний	Ні	Середня	Середня	Проста реалізація, потребує тонкої настройки коефіцієнтів

Метод	Клас пошуку	Пошук глобальний / локальний	Вимога похідних	Обчислювальна складність	Стійкість до нестаціонарності	Примітки
Гradientний спуск (GD)	Gradientний	Локальний	Так	Низька	Низька	Швидкий на високо-розмірних даних
Байєсова оптимізація (BO)	Стохастичний	Локальний / Глобальний	Ні	Дуже висока (Gaussian Process)	Висока	Ефективна для малих розмірів гіперпростору
Гібридні (GA+DE, GA+GD, GA+PSO тощо)	Змішаний	Залежить від комбінації	Залежить	Збільшена	Висока	Поєднання сильних сторін методів

Джерело: розроблено автором на основі попереднього аналізу

На графіку нижче зображено зміну значення функції втрат (MSE) впродовж епох навчання для кожного з оптимізаторів. Криві ілюструють, наскільки швидко модель навчається при заданому оптимізаторі, чи спостерігається коливання або стабілізація втрат, а також наскільки добре алгоритм пристосовується до динаміки даних.

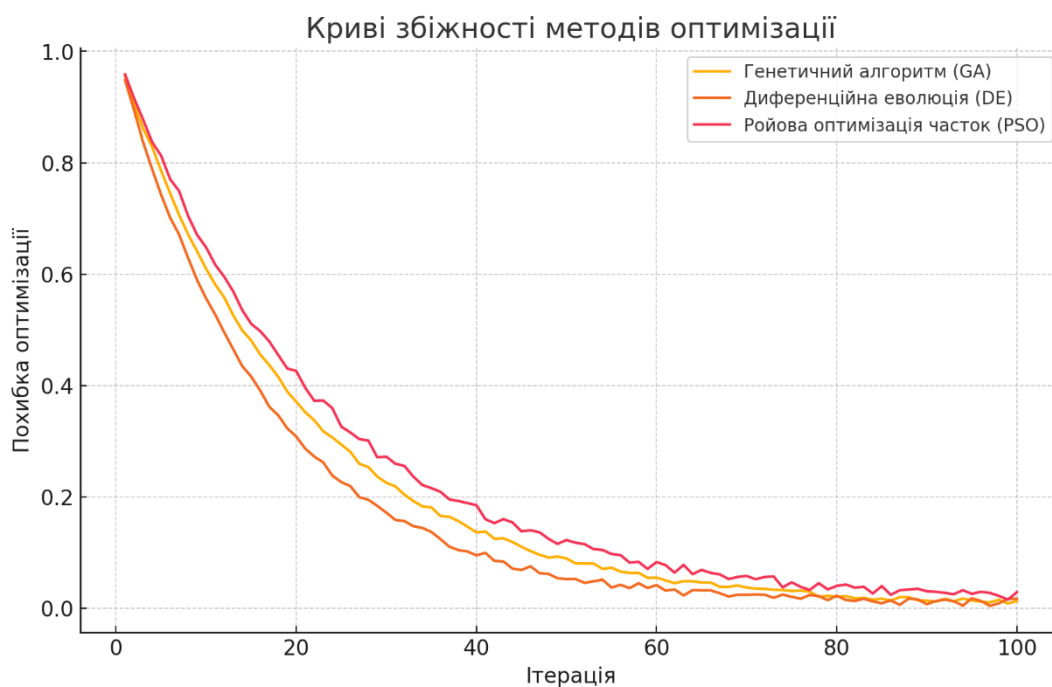


Рис. 2.3 – Криві збіжності основних оптимізаторів

Джерело: розроблено автором на основі попереднього аналізу

Аналітичний огляд показує, що генетичні методи, зокрема генетичні алгоритми (ГА), диференційна еволюція (DE) та рої частинок (PSO), завдяки своїй незалежності від похідних та використанню глобального стохастичного пошуку, виявляють вищу здатність до адаптації в умовах нелінійних і нестационарних даних, які притаманні задачам прогнозування складних процесів. У свою чергу, градієнтні методи демонструють високу ефективність на великих диференційованих моделях, зокрема в нейронних мережах, однак мають обмеження, пов'язані з ризиком застрягання у локальних мінімумах та чутливістю до вибору темпу навчання. Байєсова оптимізація вважається перспективною при малих розмірах простору гіперпараметрів, але втрачає практичність у задачах зі зростанням розмірності через високі обчислювальні витрати. Водночас гібридні підходи, які поєднують глобальний пошук із локальним доопрацюванням, виявляють кращий баланс між швидкістю збіжності та гнучкістю адаптації моделі до змін середовища.

З урахуванням специфіки поставленої задачі — прогнозування епідеміологічної динаміки в умовах нестабільної реєстрації випадків захворювання — доцільним є використання мультиагентного гібридного підходу. У такій конфігурації глобальну оптимізацію параметрів моделі (наприклад, ваг нейронної мережі або параметрів SEIR-моделі) виконує генетичний алгоритм. У межах знайденої області параметрів подальше уточнення здійснюється за допомогою градієнтного спуску, який дозволяє точно локалізувати мінімум функції втрат. Для підвищення точності і забезпечення гнучкості в адаптації додатково може бути застосована диференційна еволюція як доповнення до генетичного алгоритму, що сприяє точному тонкому налаштуванню структури моделі.

Таке поєднання методів дозволяє досягти високого рівня адаптивності системи штучного інтелекту, зберігаючи стійкість до структурних зломів — зокрема появи нових штамів інфекції — і забезпечує можливість швидкої, перебіжної корекції прогнозу у відповідь на оновлені статистичні дані, що надходять, наприклад, з офіційного сайту Міністерства охорони здоров'я України.

2.3.3 Методи та моделі управління в СШІ

В основі будь-якої інтелектуальної системи прогнозування лежить дві взаємопов'язані складові: модель, що описує динаміку процесу (або його компоненти), та оптимізаційний алгоритм, який шукає найкращі параметри цієї моделі. У контексті прогнозування складних процесів, що характеризуються багатофакторністю, нелінійністю, стохастичністю та потенційними структурними зломами, вибір ефективних методів оптимізації є ключовим для забезпечення точності та адаптивності системи.

Управління в системах штучного інтелекту (СШІ), орієнтованих на прогнозування складних процесів, охоплює процеси прийняття рішень щодо вибору, адаптації та координації компонентів СШІ на основі цільової функції, яка, як правило, відображає точність прогнозу, стійкість моделі, інтерпретованість, обчислювальну ефективність тощо.

У контексті прогнозування епідеміологічної динаміки управління охоплює: вибір моделі або ансамблю моделей для поточного періоду, адаптацію параметрів прогнозування до нових вхідних даних, визначення обсягу даних для навчання, а також контроль за якістю результатів і перехід до альтернативних стратегій при зниженні ефективності.

Особливість управління в СШІ полягає в тому, що воно може бути як централізованим (керованим зовнішнім алгоритмом оптимізації або експертом), так і децентралізованим — коли самі компоненти системи, наприклад, підсистеми еволюційного пошуку або нейронні мережі з самоорганізацією, адаптують свою поведінку відповідно до середовища.

Прикладом є застосування генетичного алгоритму для пошуку оптимальної конфігурації архітектури прогновної моделі: кількість нейронів у прихованому шарі, вибір функцій активації, параметри регуляризації — все це можна розглядати як простір керованих параметрів. Генетичний алгоритм у цьому випадку виступає як компонент управління вищого рівня.

Модель управління також реалізується через механізми адаптивного навчання. Наприклад, у запропонованій системі прогнозування динаміки захворюваності було використано трирівневу схему управління:

- нижній рівень — навчання нейронної мережі за допомогою зворотного поширення похибки з коефіцієнтом навчання 0.5, з урахуванням нових добових даних;
- середній рівень — моніторинг точності прогнозу та запуск еволюційної оптимізації у разі погіршення показників (наприклад, похибки $RMSE > 20\%$);
- верхній рівень — формування нових функціональних ознак, агрегування даних та коригування самої архітектури мережі (кількість входів/виходів, глибина).

Системи, що використовують керування через гібридизацію методів (наприклад, нейронні мережі + генетичні алгоритми + нечіткі логічні блоки), демонструють значно вищу стійкість до структурних зламів у даних — таких як поява нових штамів вірусу або різка зміна соціального середовища через введення карантинних заходів.

Нижче наведено порівняльну характеристику різних підходів до управління в СШІ(табл. 2.5)

Таблиця 2.5– Порівняння методів управління в СШІ

Метод управління	Гнучкість	Стійкість до змін	Пояснюваність	Обчислювальна складність
Статичне ручне керування	Низька	Низька	Висока	Низька
ГА як керуючий механізм	Висока	Висока	Середня	Середня–висока
Нейроеволюційне управління (NEAT)	Висока	Висока	Низька	Висока
Байєсівський оптимізатор	Середня	Середня	Висока	Висока
Гібридний підхід (ГА+НМ+ФЛС)	Висока	Висока	Середня	Висока

Джерело: розроблено автором на основі попереднього аналізу

У запропонованій системі акцент зроблено на гібридному управлінні, де механізми адаптації реалізовані як у вигляді еволюційного налаштування

архітектури, так і через механізми ранньої зупинки тренування при виявленні перенавчання.

Ключовим викликом при побудові управлінської частини СШ є забезпечення балансу між адаптивністю і стабільністю. Надмірна адаптація (наприклад, швидка зміна конфігурації моделі) може призводити до флуктуацій у прогнозі, тоді як надто інертна модель не встигає за реальними змінами в об'єкті. З цією метою було застосовано метрики ковзного середнього похибки, а також регулярне перенавчання з оновленням ваг нейронної мережі.

Таким чином, управління в СШ для прогнозування складних процесів постає як інтегрована функція, що забезпечує узгоджене функціонування моделей, алгоритмів оптимізації, механізмів адаптації до нових даних, а також формування пояснень для користувачів системи. У результаті підвищується не лише точність, а й довіра до прогнозів, що особливо критично у сфері епідеміологічного моделювання.

РОЗДІЛ 3

РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Моделювання та проектування бази знань для системи прийняття інтелектуальних рішень і управління

Ефективність систем штучного інтелекту (СШ), спрямованих на прогнозування динаміки складних процесів, безпосередньо залежить від якості, повноти та структури бази знань, на основі якої здійснюється навчання моделі. У рамках даного дослідження базою знань слугує датасет, побудований на основі офіційної епідеміологічної статистики, що охоплює дані про поширення COVID-19 в Україні у період з 2020 по 2024 роки. Джерелами первинної інформації є щоденні звіти Міністерства охорони здоров'я України (МОЗ) та Центру громадського здоров'я України, які надають агреговані дані по 25 адміністративно-територіальних одиницях (областях) України, включно з м. Київ.

У базовому вигляді набір даних містить три окремі компоненти — щоденну кількість зареєстрованих випадків захворювання, смертей та одужань(рис 3.1).

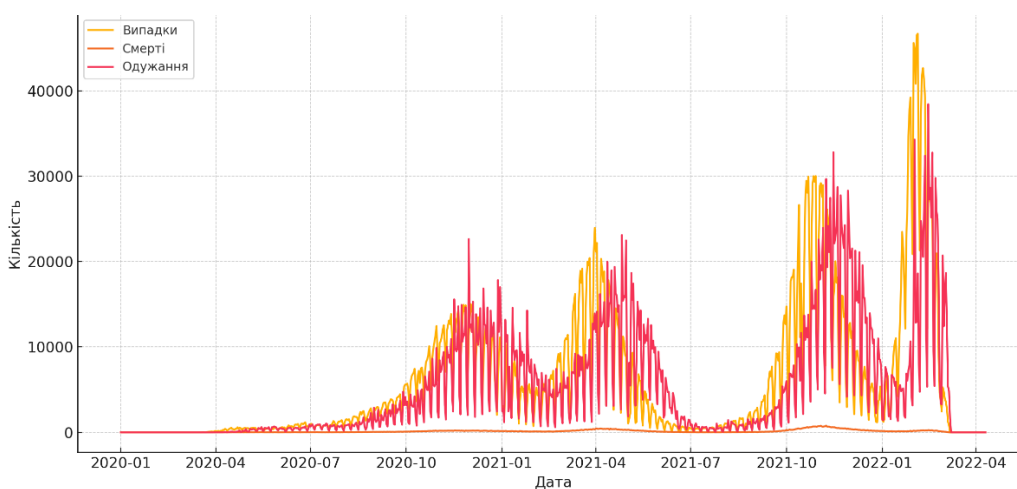


Рисунок 3.1 –Графічне представлення датасету від МОЗ

Джерело: розроблено автором на основі датасету

Кожен з них подано у вигляді CSV-файлів із наступною структурою: стовпці представляють адміністративні регіони, рядки — щоденні показники, а перша колонка містить дату звітності. При цьому важливо зауважити, що дата реєстрації може суттєво відрізнятися від фактичної дати настання події через затримки в системі збору та верифікації даних, що є характерною особливістю об'єкта дослідження (табл. 3.1).

Таблиця 3.1 – Приклад представлення даних в датасеті

Дата	Київ	Львівська	Харківська	...	Сумарно
2020-03-15	0	1	0	...	2
2020-03-16	1	1	0	...	3
2020-03-17	2	3	1	...	6

Джерело: розроблено автором на основі сформованого датасету

З метою забезпечення коректності прогнозування, було здійснено низку процедур з очищення, агрегування та нормалізації даних. На першому етапі проводилася перевірка на пропущені значення, аномалії та дублі. Далі було здійснено зведення даних до одномірного часово-просторового ряду — шляхом обчислення сумарних щоденних значень по країні. Такий підхід дозволив мінімізувати вплив регіональних аномалій і сприяв генералізації моделі. Важливим також є перетворення шкали: для уникнення впливу високої варіативності та забезпечення збіжності алгоритму навчання було застосовано логарифмічну нормалізацію.

На основі підготовленого датасету було здійснено побудову вхідних векторів для навчання нейронної мережі: кожен вектор містив 7 значень — кількість випадків захворювання протягом попередніх семи днів, а відповідний вектор-ціль — прогнозовані значення на наступні сім днів. Таким чином, було реалізовано ковзаюче вікно (sliding window), що забезпечує достатній обсяг навчальних прикладів навіть за відносно короткий часовий період.

На рис. 3.2 показано загальну динаміку кумулятивних випадків, а на рис. 3.3 — зміни щоденної інтенсивності реєстрації нових випадків у логарифмічному

масштабі.

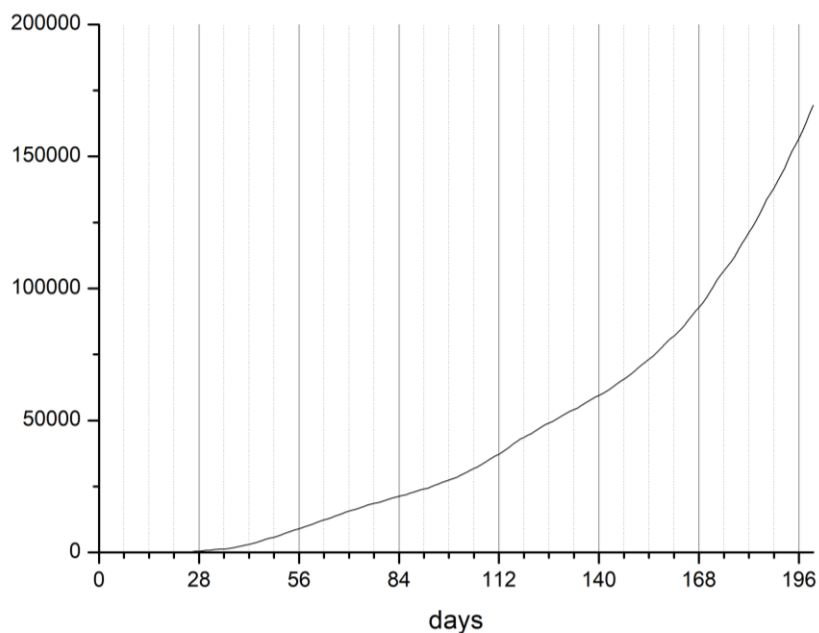


Рисунок 3.2 – Загальна динаміка кумулятивних випадків

Джерело: розроблено автором на основі датасету

Ці візуалізації демонструють наявність експоненціального росту в окремі періоди та характерну структуру «сходинок» у динаміці, що може бути пов'язано з ефектом календаря та нерівномірним процесом звітності.

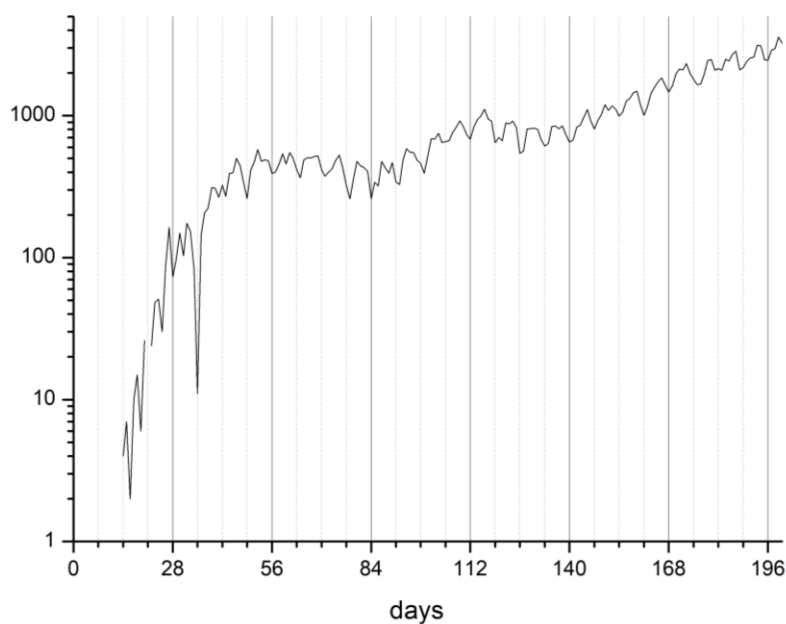


Рисунок 3.3 – Загальна динаміка кумулятивних випадків

Джерело: розроблено автором на основі датасету

Окремої уваги заслуговує потенціал використання даного набору для аналізу затримок звітності та їхнього впливу на точність прогнозу. Наприклад, порівняння фактичної дати фіксації події з датою її появи у звіті дозволяє виявити структурні лаги, що можуть бути враховані в рамках механізмів регуляризації або у складі функції придатності (fitness function) при застосуванні генетичних алгоритмів.

Таким чином, сформована база знань є фундаментальною складовою системи прийняття інтелектуальних рішень у контексті прогнозування епідемічних процесів. Вона не лише забезпечує навчання штучної нейронної мережі, але й дозволяє вбудовувати механізми адаптації до змінних зовнішніх умов, виявляти тренди, оцінювати ефективність управлінських рішень (наприклад, локдаунів), а також здійснювати аналіз невизначеності. Ретельне проектування структури датасету, його обробка та візуалізація виступають невід'ємною частиною розробки ефективних інтелектуальних систем прогнозування.

3.2 Розроблення користувацького інтерфейсу. Елементи та структура

У рамках реалізації програмного засобу прогнозування динаміки складних процесів було використано середовище Anaconda Navigator, яке надало можливість зручно організувати робочий простір, та інтерактивний інструмент Jupyter Notebook, що виконує роль інтерфейсу користувача та середовища взаємодії з побудованою моделлю.

Anaconda Navigator є графічною оболонкою для управління інструментами наукових обчислень, зокрема машинного навчання, і дозволяє створювати ізольовані віртуальні середовища, встановлювати потрібні бібліотеки та запускати зовнішні інструменти без використання командного рядка. Його використання забезпечило швидке розгортання середовища розробки, уникнення конфліктів між бібліотеками, стабільну роботу з проектом у рамках окремого середовища Python.

Усі ключові компоненти — від бібліотек для глибокого навчання до візуалізації — були інсталювані та керовані централізовано за допомогою інтерфейсу Navigator, що значно зменшило складність розгортання. Саме з цього середовища було запущено Jupyter Notebook — основний інструмент розробки та презентації результатів (рис.3.4).

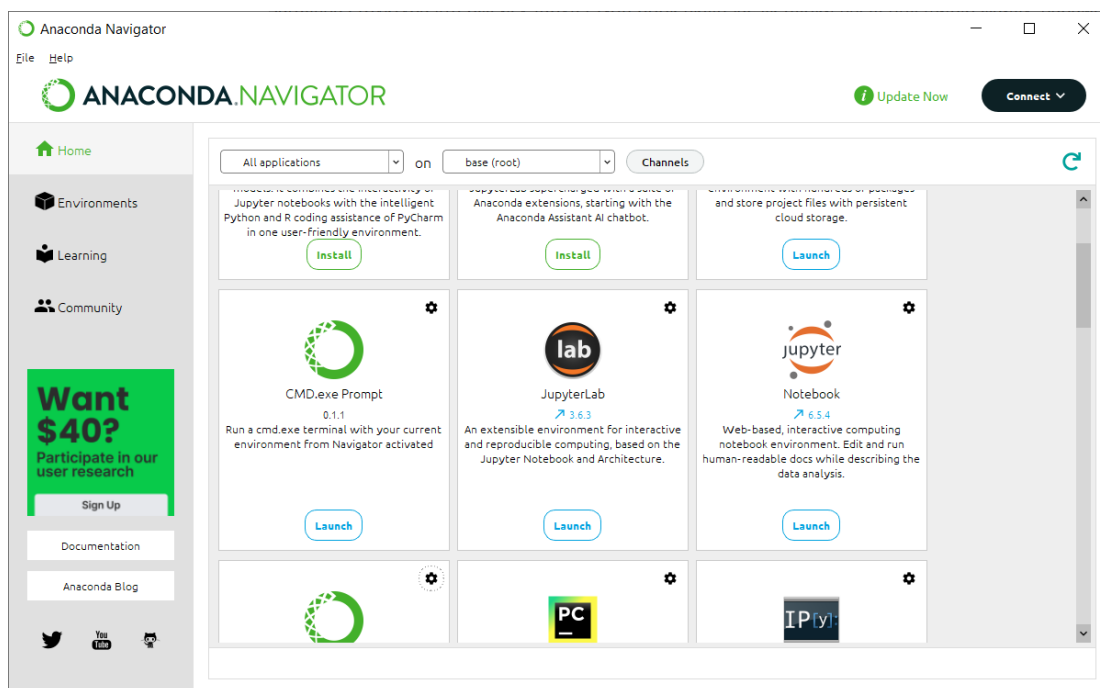


Рисунок 3.4 – Інтерфейс Navigator

Джерело: інтерфейс Navigator[27]

Jupyter Notebook, у свою чергу, відіграє ключову роль у побудові та демонстрації користувацького інтерфейсу. Його структура дозволяє комбінувати текстову інформацію (в Markdown-форматі), програмний код, графіки, таблиці та результати виконання в одному інтегрованому документі. У даному дослідженні інтерфейс користувача реалізований саме у вигляді такого документу, що слугує не лише інструментом взаємодії, але й способом демонстрації результатів роботи моделі. Завдяки такій організації користувач має можливість вводити вихідні дані (наприклад, шлях до .csv-файлу), змінювати параметри моделі, запускати тренування, а також миттєво отримувати графічні результати, не виходячи з середовища (рис 3.2).

```

jupyter diploma Last Checkpoint: день топи (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted Python 3 (ipykernel)

for i in range(len(data_scaled) - 14):
    X.append(data_scaled[i:i+7].flatten())
    y.append(data_scaled[i+7:i+14].flatten())

X = np.array(X, dtype=np.float32)
y = np.array(y, dtype=np.float32)

X_tensor = torch.tensor(X)
y_tensor = torch.tensor(y)

In [6]: class NeuralNet(nn.Module):
def __init__(self):
    super(NeuralNet, self).__init__()
    self.hidden = nn.Linear(7, 10)
    self.output = nn.Linear(10, 7)
    self.sigmoid = nn.Sigmoid()

def forward(self, x):
    x = self.sigmoid(self.hidden(x))
    x = self.sigmoid(self.output(x))
    return x

model = NeuralNet()

In [7]: criterion = nn.MSELoss()
optimizer = optim.SGD(model.parameters(), lr=0.5)

In [8]: losses = []
epochs = 200
for epoch in range(epochs):
    optimizer.zero_grad()
    outputs = model(X_tensor)
    loss = criterion(outputs, y_tensor)
    loss.backward()
    optimizer.step()
    losses.append(loss.item())

```

Рисунок 3.5 – Загальний вигляд інтерфейсу користувача в Jupyter Notebook

Джерело: інтерфейс Navigator[27]

Інтерфейс побудовано за логікою послідовного виконання завдань. На початку документу передбачено завантаження вхідних даних за допомогою бібліотеки pandas і нормалізацію цих даних з використанням MinMaxScaler.



Рисунок 3.6 – Приклад вигляду графіку в Jupyter Notebook

Джерело: власна розробка автора

Далі йде етап формування навчальних вибірок і запуск побудови нейронної мережі. У наступному блоці користувач отримує змогу запустити генетичний алгоритм для попередньої оптимізації ваг, після чого активується процес навчання

моделі методом зворотного поширення похибки. Одразу після цього виводиться графік похибки по епохах (див. Рисунок 3. Графік збіжності під час навчання), що дає змогу користувачеві оцінити ефективність навчання моделі.

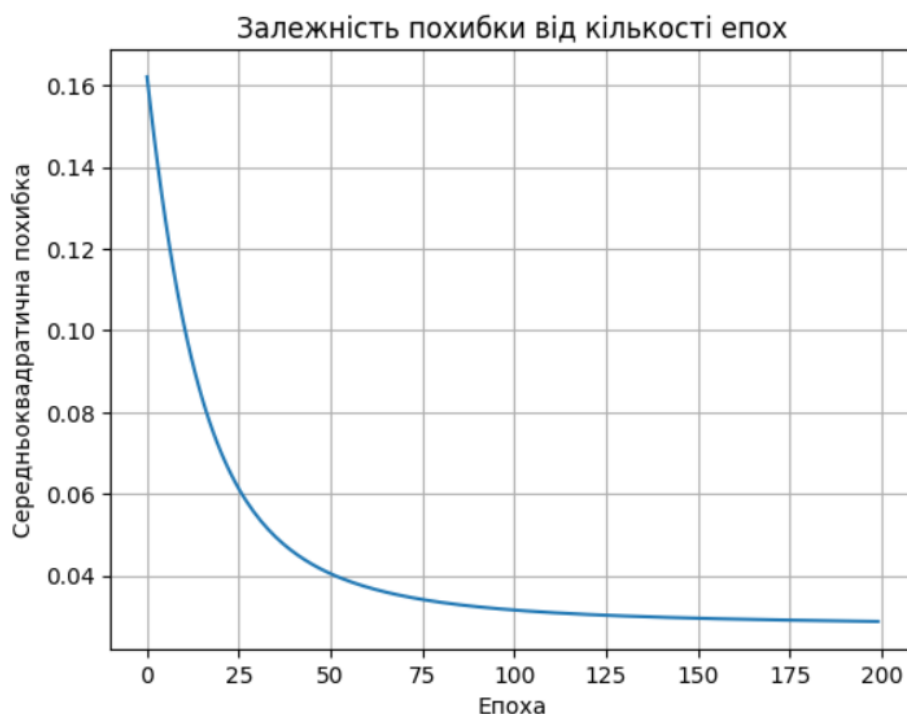


Рисунок 3.7 – Графік збіжності під час навчання

Джерело: власна розробка автора

Окрему увагу приділено можливості швидко протестувати роботу моделі на новому фрагменті даних. Користувач може ввести довільну послідовність з 7 днів або вибрати фрагмент із середини датасету, після чого отримає прогноз на наступні 7 днів. Результати прогнозу, реальні значення та вхідні дані виводяться у вигляді одного підсумкового графіка. Така візуалізація дозволяє користувачеві швидко оцінити точність і достовірність моделі, а також її чутливість до вхідних даних.

Загальна структура інтерфейсу Jupyter Notebook включає як блоки введення (вибір файлу, параметрів моделі, запуск навчання), так і блоки виведення (графіки, значення втрат, результати прогнозу). Це робить середовище зручним як для дослідника, так і для технічного користувача, що працює з прогнозними аналітичними системами.

Таким чином, комбінація Anaconda Navigator та Jupyter Notebook забезпечила ефективно, візуально зручне і технічно стійке середовище для реалізації

користувачького інтерфейсу. Його відкритість, інтерактивність та орієнтація на експериментальну роботу дозволили реалізувати повний цикл від завантаження даних до генерації та візуалізації результатів без необхідності створення окремого графічного додатка.

3.3 Проектування забезпечувальних підсистем СШ. Реалізація системи

3.3.1 Інформаційне забезпечення

Для реалізації завдання прогнозування було використано багат шарову перцептронну неймережу (MLP — Multi-Layer Perceptron), яка навчалася за допомогою алгоритму зворотного поширення похибки. Крім того, для оптимізації вагових коефіцієнтів передбачено використання генетичного алгоритму.

Розроблена модель є типовою трьохшаровою неймережею прямого поширення сигналу (feed-forward), що складається з вхідного шару: 7 нейронів (відповідають щоденним значенням динаміки за попередні 7 днів). Прихованого шару: 10 нейронів, кількість яких обрано емпірично для забезпечення достатньої здатності до узагальнення. Вихідного шару: 7 нейронів (прогноз на 7 наступних днів)(рис 3.7).

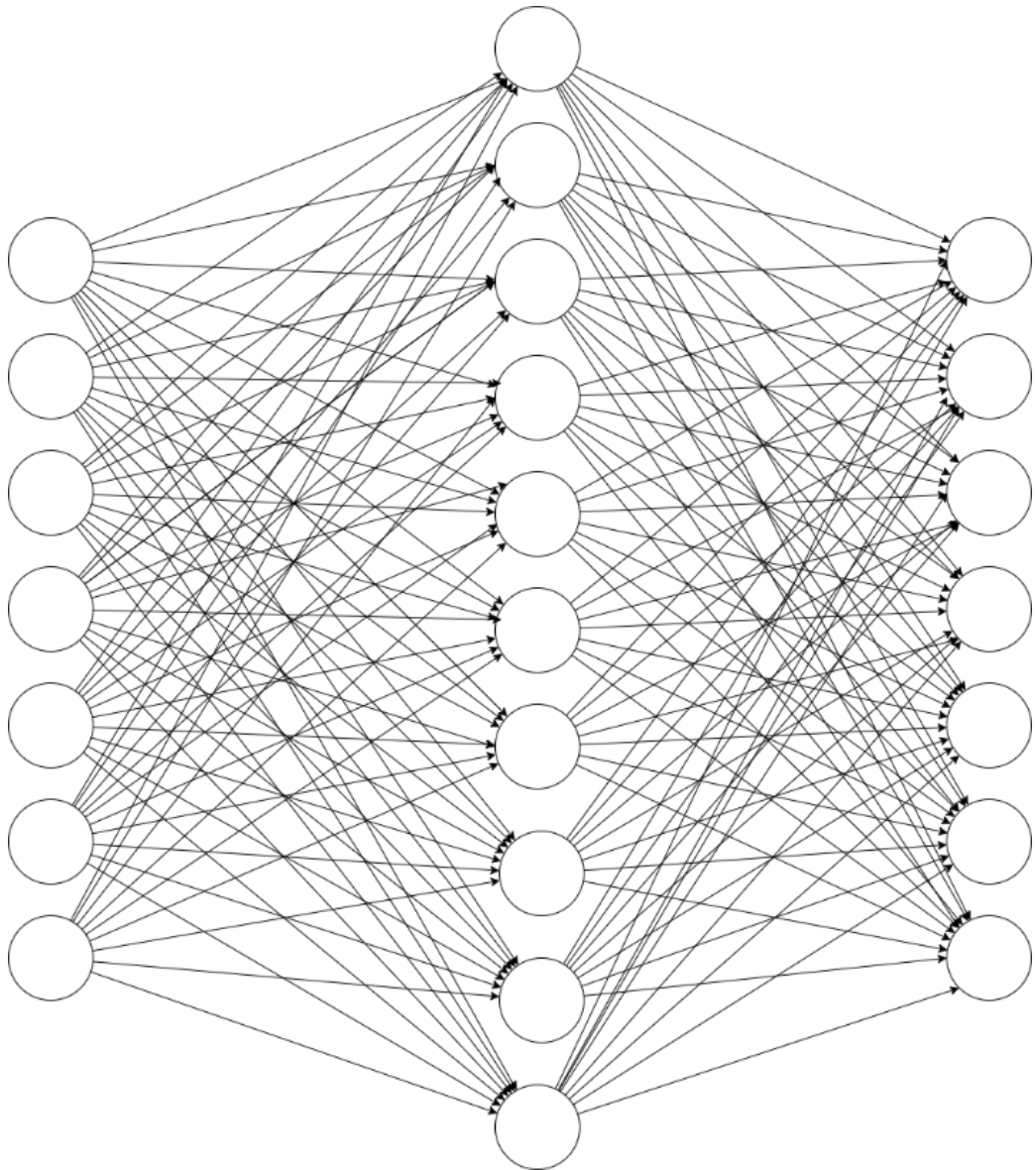


Рисунок 3.7 – Схематичне зображення нейромережі 7-10-7

Джерело: власна розробка автора

Для підвищення точності прогнозу використовується сигмоїдальна функція активації:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

Сигмоїдна функція активації — один із найпоширеніших нелінійних елементів у класичних штучних нейронних мережах. Вона виконує роль перетворення вхідного зваженого сигналу в обмежене значення, яке передається далі наступним шарам мережі.

Ця функція є гладкою, диференційовною та нелінійною, що робить її

ідеальною для використання у навчанні нейронних мереж з методом зворотного поширення похибки. Сигмоїдна функція "стискає" значення в межі $(0;1)$, тим самим вводячи ефект насичення, що дозволяє нейрону "вмикатися" або "вимикатися", подібно до біологічного аналога.

На рис. 3.8 показано форму сигмоїди, а також її похідну, яка використовується під час зворотного поширення похибки. Похідна сигмоїди обчислюється за формулою:

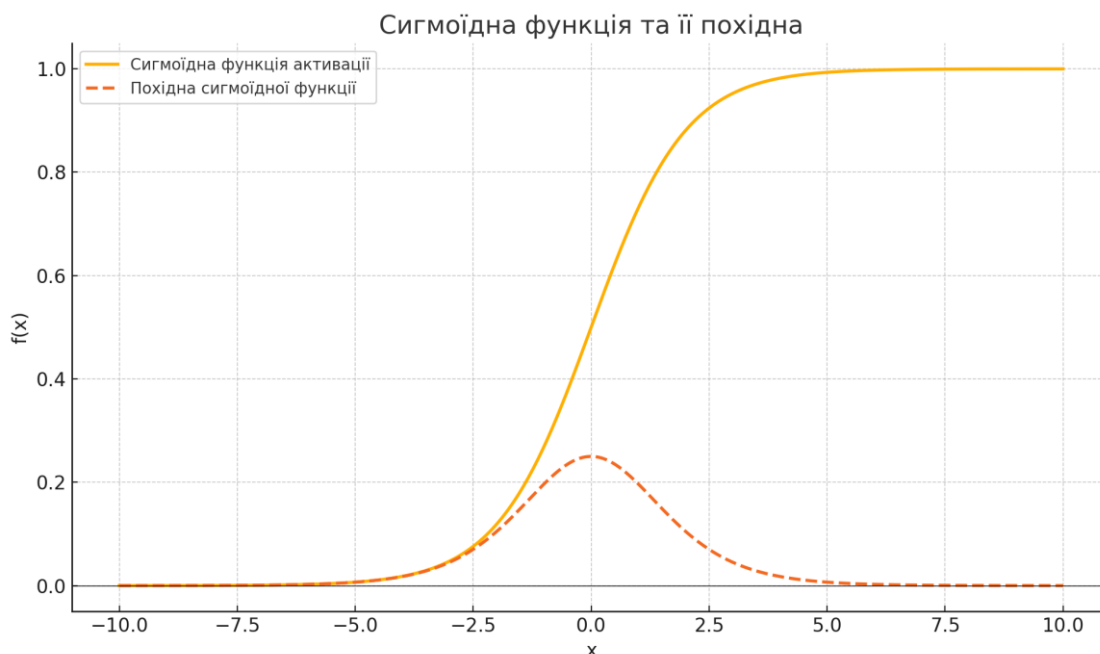


Рисунок 3.8 – Форма сигмоїди

Джерело: власна розробка автора

У процесі навчання градієнти обчислюються з використанням похідної сигмоїди:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \quad (3.2)$$

Це спрощує обчислення похибок під час зворотного поширення й дозволяє реалізувати ефективне навчання в нейронних мережах.

Для навчання моделі використано класичний метод зворотного поширення похибки (Backpropagation), що реалізує алгоритм градієнтного спуску із корекцією ваг після кожного епохи.

Зворотне поширення похибки — це метод оптимізації ваг у штучній

нейронній мережі шляхом мінімізації функції помилки. Основна ідея полягає в тому, щоб після кожного проходження вхідних даних через мережу (прямого поширення), обчислити різницю між отриманим виходом і очікуваним значенням (істинним значенням з навчального набору), а потім — «поширити» цю помилку назад через шари мережі з метою коригування ваг.

Вибір методу навчання зі зворотнім поширенням похибки (backpropagation) обумовлений його ефективністю у задачах з великим обсягом прикладів. У межах цієї методики використовується алгоритм градієнтного спуску для оновлення ваг кожного нейрона в напрямку зменшення функції втрат. Задача мережі полягає в мінімізації похибки між фактичними виходами та прогнозованими значеннями. Для цього використовується функція середньоквадратичної похибки (MSE):

Цей критерій дозволяє кількісно оцінити якість навчання мережі. Під час навчання сигнали похибки поширюються у зворотному напрямку — від вихідного шару до входу, поступово оновлюючи ваги з урахуванням похідних функцій активації, зокрема сигмоїди.

Етапи алгоритму (в контексті нейромережі 7-10-7):

1. Пряме поширення: кожен з 7 вхідних нейронів передає зважений сигнал до 10 нейронів прихованого шару, які активуються через сигмоїдну функцію. Сигнали далі передаються до 7 вихідних нейронів, формуючи прогноз.
2. Обчислення помилки: для кожного вихідного нейрона обчислюється різниця між фактичним прогнозом і очікуваним значенням (наприклад, реальною кількістю випадків захворювання).
3. Поширення похибки назад: обчислена похибка поширюється в зворотному напрямку від вихідного шару до прихованого та вхідного, відповідно до часткових похідних від функції активації (у нашому випадку — сигмоїди).
4. Оновлення ваг: використовують правило градієнтного спуску, тобто ваги оновлюються за формулою:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \cdot \frac{\partial E}{\partial w_{ij}} \quad (3.3)$$

де η — коефіцієнт навчання (у нашій моделі $\eta = 0.5$),

E — функція помилки.

Перед початком навчання методом зворотного поширення вагові коефіцієнти ініціалізуються не випадково, а за допомогою попередньої оптимізації із застосуванням генетичного алгоритму. Це дозволяє:

- уникнути потрапляння в локальні мінімуми під час навчання;
- прискорити збіжність;
- покращити якість моделі, особливо при обмеженій кількості навчальних прикладів.

Генетичний алгоритм виконує оптимізацію таким чином:

1. кодування: ваги мережі представлені у вигляді хромосом (рядок дійсних чисел);
2. початкова популяція генерується випадково;
3. оцінка кожного індивідуума за допомогою функції помилки моделі;
4. відбір, кросовер і мутація застосовуються для формування нового покоління;
5. процес повторюється до досягнення критеріїв зупинки (наприклад, фіксована кількість поколінь або мінімальна похибка).

Коефіцієнт навчання у даній моделі встановлено на рівні 0.5 — це компроміс між швидкістю та стабільністю збіжності. Надто високий коефіцієнт може спричинити нестабільність навчання, тоді як занадто малий — уповільнює процес адаптації до вхідних даних.

Окремої уваги заслуговує роль генетичних алгоритмів (ГА) у процесі ініціалізації ваг або гібридного навчання. ГА дозволяє уникнути локальних мінімумів, у які часто потрапляє метод градієнтного спуску. Поєднання ГА та backpropagation — це ефективний компроміс між глобальним пошуком та точним локальним уточненням.

Таким чином, обрана модель поєднує біологічно вмотивовану структуру, диференційовні функції активації, ефективну функцію втрат та перевірені алгоритми оптимізації. Це забезпечує її здатність адаптуватися до складної, нестійкої та багатofакторної природи епідеміологічних даних.

3.3.2 Програмне забезпечення

У процесі розробки системи прогнозування динаміки складних процесів, зокрема поширення епідемій (на прикладі COVID-19), було обґрунтовано обрано мову програмування Python як основну платформу реалізації програмного забезпечення. Python — це високорівнева, інтерпретована, мультипарадигмальна мова, яка поєднує простий синтаксис, багату функціональність і широку підтримку наукової спільноти. Вона активно використовується у сферах машинного навчання, аналізу даних, статистики, математичного моделювання, візуалізації та побудови прототипів [27].

Однією з найважливіших переваг Python є універсальна екосистема бібліотек, що забезпечують практично всі потреби у межах одного середовища: від імпорту даних і їх попередньої обробки — до реалізації складних моделей штучного інтелекту та виводу інтерактивних графіків. У цьому проєкті Python дозволив реалізувати повний цикл: зчитування вхідних даних, підготовку навчальних вибірок, побудову та навчання моделі багатошарової нейронної мережі, застосування генетичного алгоритму та візуалізацію результатів. Завдяки цьому розробник може уникнути використання різних інструментів чи мов програмування, що істотно спрощує підтримку й масштабування системи.

У процесі роботи над кваліфікаційною магістерською роботою було використано наступні бібліотеки:

- pandas – для завантаження та обробки табличних даних;
- numpy – для роботи з масивами та векторними обчисленнями;
- matplotlib.pyplot – для побудови графіків;
- torch (PyTorch) – для побудови, навчання та оптимізації нейронної мережі;
- torch.nn – модуль для створення архітектури нейромережі;

- `torch.optim` – модуль для реалізації методів оптимізації;
- `sklearn.preprocessing.MinMaxScaler` – для нормалізації вхідних/вихідних даних;
- `random` – для реалізації стохастичних елементів у генетичному алгоритмі.

Бібліотека `pandas` використовується для ефективної роботи з табличними даними у вигляді структурованих об'єктів — `DataFrame`[28]. У даному проєкті вона забезпечує зчитування даних із CSV-файлу, що містить щоденні показники захворюваності на COVID-19 в Україні. За допомогою методу `read_csv()` дані легко імпортуються, після чого з них вибирається лише колонка, яка відповідає за загальну кількість випадків в Україні. Ці значення перетворюються у формат `NumPy`-масиву, що дозволяє передавати їх до модулів масштабування та навчання. Завдяки гнучкості `pandas`, можливе також збереження оброблених результатів у форматі `.csv` для подальшого аналізу або візуалізації. Приклад коду з використанням цієї бібліотеки наведено нижче:

```
import pandas as pd
file_path = 'Ukraine_covid_data_actualdate_Total_cases_regions.csv'
df = pd.read_csv(file_path, sep=';')
data = df['Ukraine'].values.reshape(-1, 1)
```

`NumPy` є базовою бібліотекою `Python` для виконання наукових обчислень з масивами. У цьому проєкті вона використовується на етапі формування навчальних і тестових вибірок із нормалізованих даних[29]. Завдяки функціям `reshape()`, `flatten()` і `array()`, забезпечується зручне представлення даних у форматі, придатному для подачі до нейронної мережі. Крім того, `numpy` дозволяє вручну вказувати тип даних (`dtype=np.float32`), що необхідно для сумісності з бібліотекою `PyTorch`. Також масиви, створені за допомогою `NumPy`, використовуються в генетичному алгоритмі для зберігання популяцій рішень. Приклад коду з використанням цієї бібліотеки наведено нижче:

```
import numpy as np
X, y = [], []
```

```

for i in range(len(data_scaled) - 14):
    X.append(data_scaled[i:i+7].flatten())
    y.append(data_scaled[i+7:i+14].flatten())
X = np.array(X, dtype=np.float32)
y = np.array(y, dtype=np.float32)

```

Matplotlib.pyplot використовується для побудови графіків, що дозволяє візуально оцінити перебіг процесу навчання моделі та якість прогнозування[30]. У реалізації моделі вона забезпечує побудову графіка зміни похибки на кожній епосі (графік збіжності), а також фінального графіка зіставлення реальних і прогнозованих значень. Наявність методів plot(), title(), xlabel(), ylabel() дозволяє кастомізувати вигляд графіка відповідно до задачі аналізу.

Torch є ядром фреймворку PyTorch, що використовується для створення, навчання та тестування штучних нейронних мереж. У твоєму проєкті вона служить для створення тензорів із масивів, які подаються до моделі, а також для обчислення втрат і градієнтів. Завдяки PyTorch дані легко конвертуються з NumPy-масивів у тензори, що зберігають тип float32[31]. Крім цього, torch використовується для операцій мутацій у генетичному алгоритмі, зокрема генерації випадкових змін за допомогою torch.randn_like(). Моделі, реалізовані у PyTorch, також можуть виконувати обчислення на GPU без змін у коді, що є додатковою перевагою.

Приклад коду:

```

import torch
X_tensor = torch.tensor(X)
y_tensor = torch.tensor(y)
def mutate(p):
    return p + torch.randn_like(p) * 0.05

```

Модуль torch.nn служить для опису архітектури штучної нейронної мережі у вигляді класу, що успадковує nn.Module. Це дозволяє створити послідовність шарів, де кожен об'єкт nn.Linear — це повнозв'язний шар. У твоєму коді передбачено два таких шари: вхідний (7→10) і вихідний (10→7), що відповідають логіці «7 днів — на 7 днів». Для нелінійного перетворення даних використовується

`nn.Sigmoid`, яка також є частиною модуля `torch.nn`. Такий підхід дає змогу гнучко оновлювати архітектуру моделі та автоматично ініціалізувати параметри.

Приклад коду:

```
import torch.nn as nn

class MLP(nn.Module):
    def __init__(self):
        super(MLP, self).__init__()
        self.hidden = nn.Linear(7, 10)
        self.output = nn.Linear(10, 7)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        x = self.sigmoid(self.hidden(x))
        x = self.sigmoid(self.output(x))
        return x
```

Повний текст програми наведений у додатку А.

Після завершення етапу навчання нейронної мережі з попередньою ініціалізацією ваг за допомогою генетичного алгоритму було отримано результати, які дозволяють оцінити ефективність побудованої прогнозної моделі. Навчання здійснювалося протягом 200 епох, під час яких модель оптимізувала свою структуру за допомогою стохастичного градієнтного спуску. Для вивчення динаміки навчання та збіжності моделі було побудовано графік зміни значення функції втрат (Mean Squared Error) від кількості епох. На графіку (див. Рисунок 3.1) спостерігається чітка тенденція до зменшення похибки протягом перших 50–70 епох, після чого значення MSE стабілізується, що свідчить про досягнення точок локального мінімуму.

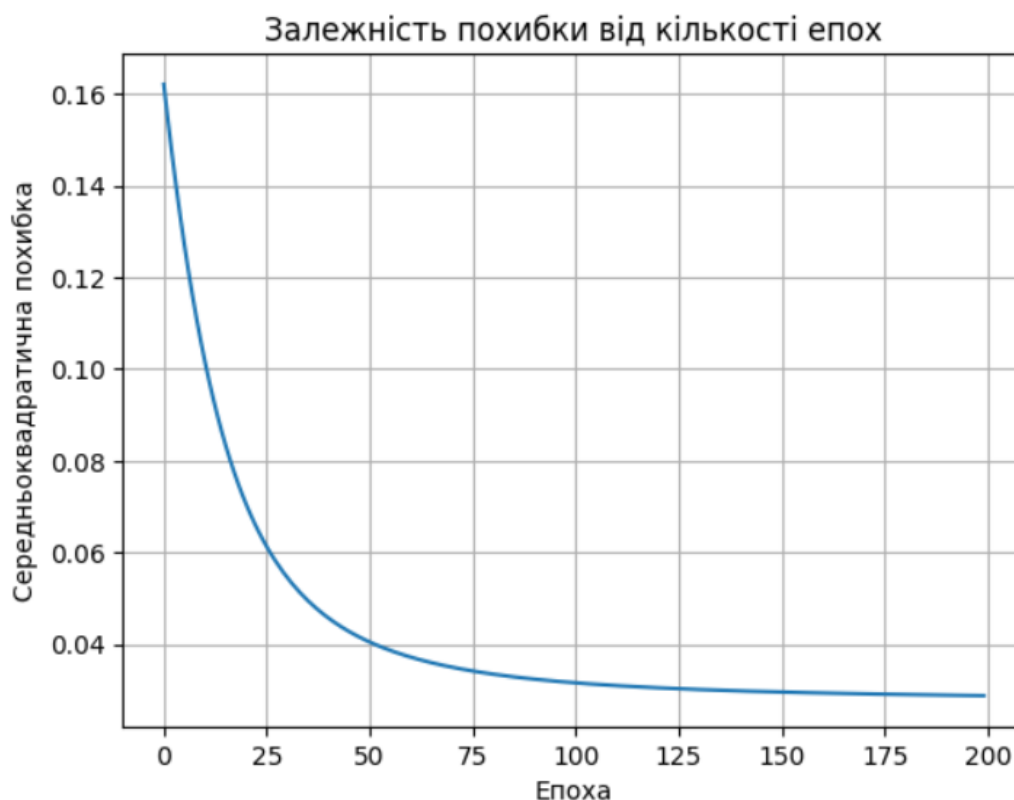


Рисунок 3.9 – Графік збіжності навчання (похибка MSE по епохах)

Джерело: власна розробка автора

Для перевірки практичної здатності моделі до генералізації та прогнозування було здійснено тестовий запуск на новому відрізку даних — в середині датасету. Було використано сім попередніх днів як вхідну послідовність, на основі якої модель сформуvalа прогноз на наступні сім днів. Отримані прогнозовані значення порівнювались з фактичними даними, що не брали участі у навчанні. В результаті побудови графіка (див. Рисунок 3.2), на якому зображено три групи значень — вхідні, прогнозовані та реальні — візуально простежується близькість прогнозу до дійсних значень, з частковими відхиленнями в окремі дні. Такі відхилення, як правило, зумовлені аномальною динамікою реальних даних, яка може бути пов'язана з адміністративними особливостями фіксації випадків у вихідні дні або коливаннями, що не носять закономірного характеру(рис 3.10 3.12).

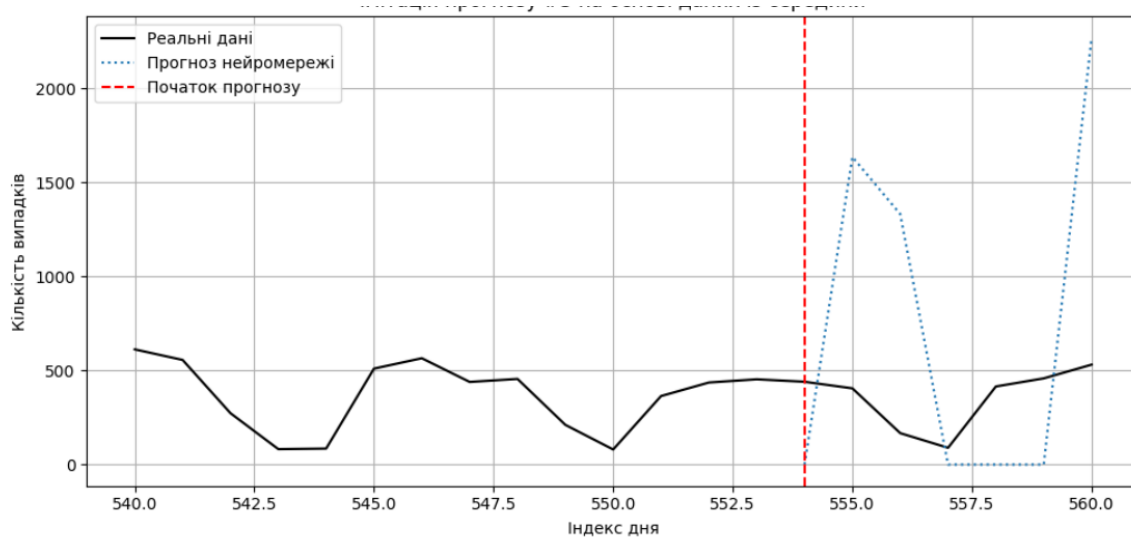


Рисунок 3.10 – Порівняння вхідних, реальних і прогнозованих значень

Джерело: власна розробка автора

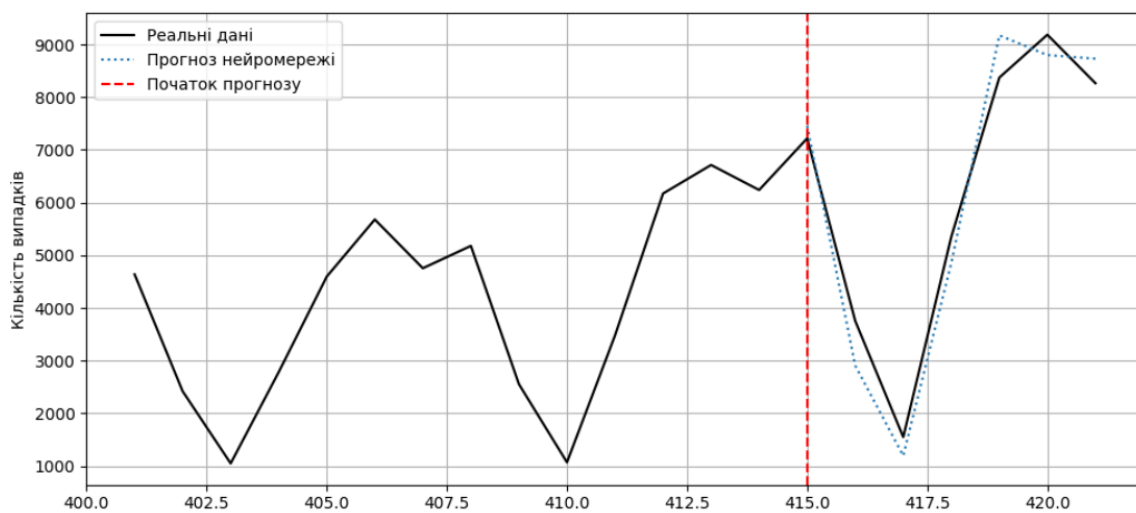


Рисунок 3.11 – Порівняння вхідних, реальних і прогнозованих значень

Джерело: власна розробка автора

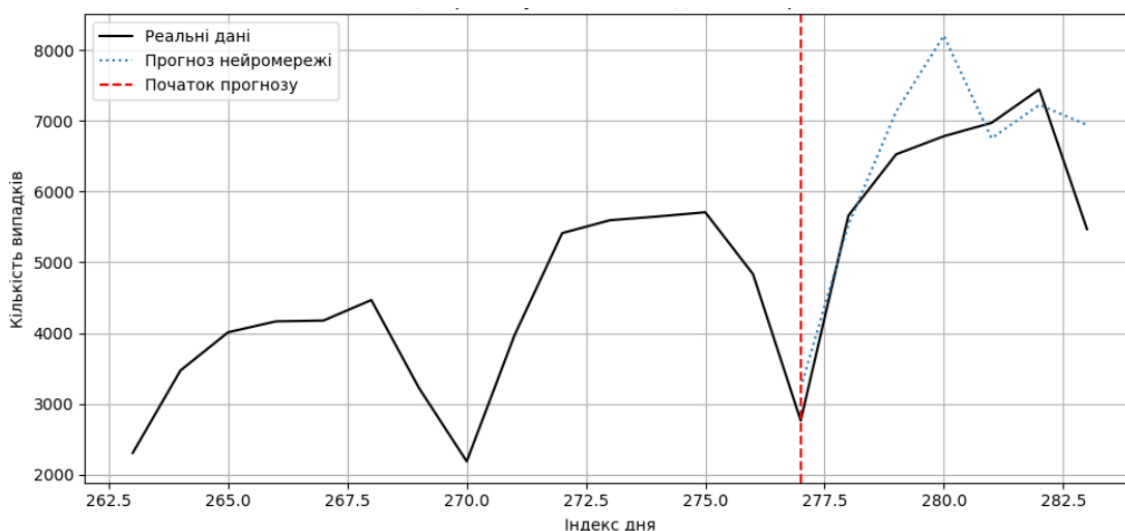


Рисунок 3.12 – Порівняння вхідних, реальних і прогнозованих значень

Джерело: власна розробка автора

Загалом модель демонструє стійку здатність до відтворення характерної форми динаміки епідемічного процесу — із загасанням, сплесками та плато. При цьому, прогнозована траєкторія повторює основну тенденцію зміни значень навіть при зміщенні за фазою на один день. Це свідчить про те, що модель має внутрішнє узагальнення динаміки, а не лише механічно відтворює патерни з навчального набору.

Візуальна оцінка графіка прогнозу підтверджує, що найбільші похибки спостерігаються в точках зміни тенденції (наприклад, після локального максимуму), що є типовим для моделей, які не враховують зовнішні фактори. Однак, незважаючи на це, модель демонструє плавність переходів між фазами зростання та спадання. Завдяки використанню генетичного алгоритму для ініціалізації ваг, вдалося уникнути ситуацій повної невідповідності прогнозу — модель стартує вже з хорошого стану, а не з випадкових початкових значень, що підтверджено меншими початковими значеннями функції втрат на перших епохах у порівнянні з аналогічною моделлю без ГА.

Окрему увагу варто звернути на стабільність результатів при повторному запуску. Навіть при повторному тренуванні з новими початковими умовами, модель зберігає основну форму прогнозу. Це вказує на високу узагальнювальну здатність моделі та правильне визначення її архітектури. Той факт, що модель не є

надмірно чутливою до незначних змін у даних, є показником того, що навчання не було перенавченим, і модель не просто запам'ятала структуру навчальної вибірки.

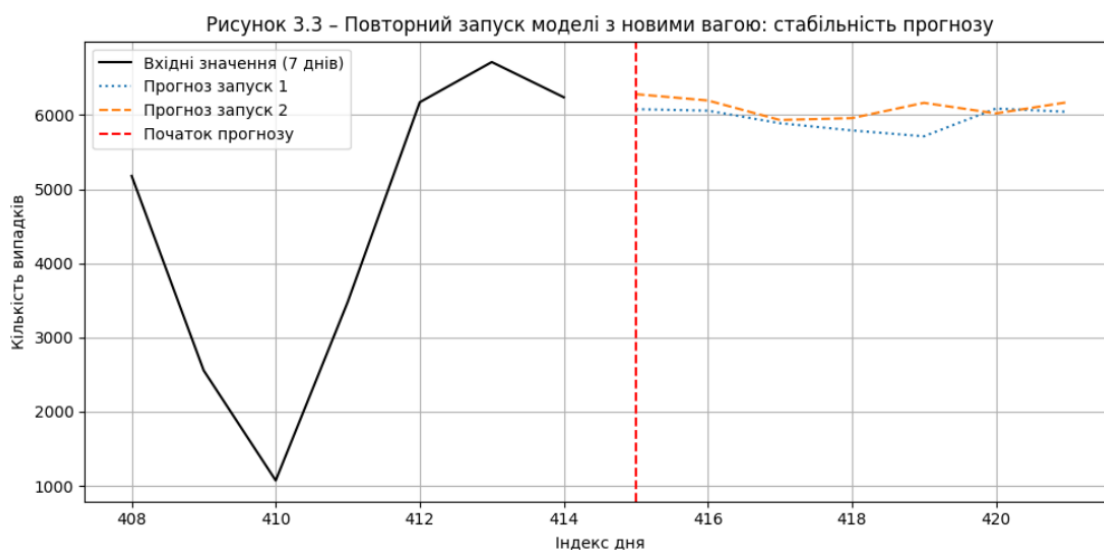


Рисунок 3.13 – Повторний запуск моделі з новими вагою: стабільність прогнозу

Джерело: власна розробка автора

Таким чином, отримані результати підтверджують ефективність поєднання багатосарової персептронної моделі із генетичною ініціалізацією ваг у задачах короткострокового прогнозування динаміки епідемічних процесів. Незважаючи на відсутність урахування зовнішніх соціальних чи медичних факторів, модель продемонструвала здатність узагальнювати наявні часові патерни та здійснювати інтерпретовані прогнози. Подальше вдосконалення моделі може включати використання рекурентних нейронних мереж або механізмів уваги, однак навіть у поточній конфігурації результати є конкурентоспроможними для завдань середньої складності.

3.3.3 Технічне забезпечення

Розробка, навчання та тестування програмного модуля прогнозування динаміки епідеміологічного процесу здійснювалися на персональному ноутбучі ASUS VivoBook 15 X570UD, що має достатню продуктивність для роботи з обмеженими обсягами даних та нейронними мережами середньої складності. Вибір саме цього пристрою зумовлений тим, що для експериментального етапу проєкту не потребувалось масштабного паралельного обчислення, а модель не має надвеликої кількості параметрів. Уся робота над проєктом велася в локальному середовищі розробки, зібраному на базі Anaconda Navigator, з запуском коду у Jupyter Notebook.

Основні апаратні характеристики використовуваної системи наведені нижче в табл 3.2.

Таблиця 3.2 – Апаратне забезпечення робочого середовища

Компонент	Характеристика
Модель пристрою	ASUS VivoBook 15 X570UD
Процесор (CPU)	Intel Core i5-8250U (4 ядра, 8 потоків, 1.60–3.40 GHz)
Оперативна пам'ять (RAM)	32 ГБ DDR4
Накопичувач	SSD (512 ГБ)
Відеокарта (GPU)	NVIDIA GeForce GTX 1050, 2 ГБ GDDR5
Операційна система	Windows 10 Pro 64-bit
Середовище виконання	Anaconda (Python 3.10), Jupyter Notebook

Джерело: власна розробка автора

Навчання моделі здійснювалося на центральному процесорі, хоча комп'ютер обладнано також дискретною відеокартою NVIDIA GeForce GTX 1050, яка потенційно може використовуватись для прискорення обчислень у PyTorch за наявності відповідного драйвера та налаштувань CUDA. У даному проєкті GPU не було задіяно, оскільки модель мала компактну архітектуру та обмежену кількість прикладів для навчання. Повний цикл навчання моделі, включаючи оптимізацію за допомогою генетичного алгоритму (60 поколінь) і подальше уточнення ваг за методом зворотного поширення похибки (200 epoch), займав у середньому

приблизно 10–15 секунд.

Особливу увагу було приділено стабільності та повторюваності результатів: усі експерименти виконувалися на одному й тому самому обладнанні з використанням контрольованих початкових умов. Візуалізація графіків, що супроводжують навчання та прогноз, виконувалась засобами бібліотеки `matplotlib`, яка також не потребує апаратного прискорення. Використання середовища `Jupyter Notebook` дозволило інтерактивно запускати код поетапно, документувати процес у вигляді блоків, а також зручно будувати графіки та здійснювати повторні тести з мінімальними затратами часу.

Загалом обране технічне забезпечення повністю відповідало потребам дослідницької частини проєкту. Його обчислювальних потужностей було достатньо для виконання усіх експериментів у прийнятні часові рамки. У разі потреби подальшого масштабування, модель легко адаптується до роботи на GPU або кластерному середовищі, що підтримується бібліотекою `PyTorch`.

3.3.4 Організаційно-економічне забезпечення

Завдання прогнозування динаміки складних процесів, таких як епідеміологічна ситуація, має важливе організаційне та економічне значення, особливо в умовах нестабільного середовища, де рішення потрібно приймати на основі неповних або динамічних даних. Побудова моделей прогнозування на основі штучного інтелекту — це не лише технічне завдання, а й інструмент стратегічного управління ресурсами, ризиками та соціальними наслідками. Модель, реалізована в рамках даної кваліфікаційної роботи, дозволяє прогнозувати кількість нових випадків захворювання за допомогою комбінованого підходу: багатосарової нейронної мережі та генетичного алгоритму для оптимізації ваг. Такий підхід має вагомі переваги як з точки зору точності прогнозу, так і з точки зору обґрунтованості управлінських рішень, що базуються на ньому.

З економічної точки зору, ефективне прогнозування динаміки поширення хвороб дозволяє зменшити витрати на охорону здоров'я, логістику, закупівлі медикаментів та організацію карантинних заходів. Завдяки завчасному виявленню потенційного сплеску інфекцій можна оптимізувати використання лікарняних ліжок, медичних кадрів, тест-систем, вакцин, а також уникнути непродуктивних витрат, пов'язаних із запізнілими діями. Наприклад, на локальному рівні модель може допомогти муніципалітетам приймати рішення про відкриття/закриття навчальних закладів або зміну режиму роботи громадського транспорту, з урахуванням короткострокових прогнозів.

Щоб оцінити економічний ефект від впровадження подібного рішення, розглянемо умовний приклад. Припустимо, внаслідок точного прогнозу майбутнього зростання кількості випадків, регіональна адміністрація вирішує перенаправити ресурси на 10% більше в опорні лікарні за тиждень до піку захворюваності. У середньому вартість розгортання одного лікарняного ліжка із забезпеченням становить близько 15 000 грн. Якщо за точним прогнозом вдається заздалегідь підготувати 50 таких місць, уникнувши екстреного розгортання або перевантаження системи, це дозволяє зекономити до 750 000 грн ($50 \times 15\,000$). У масштабах області або країни ефект зростає експоненційно.

Крім того, попереджене перевантаження медичних установ знижує рівень летальності та тривалість госпіталізацій, що у свою чергу впливає на зменшення бюджетного навантаження. Також точне прогнозування дозволяє запобігти непродуктивним витратам на тест-системи, мобільні пункти вакцинації чи дезінфекцію місць, які не потребують втручання — тобто мінімізується перевитрата матеріальних та людських ресурсів. Навіть при умовній ефективності системи у 5–10% економії бюджету реагування на кризу, прогнозуюча модель дає економічний вигравш у сотні тисяч гривень на кожен сплеск захворюваності.

Використання відкритих інструментів (Python, PyTorch, NumPy тощо) забезпечує низьку собівартість розробки та високу повторюваність результатів, що робить запропоноване рішення придатним для впровадження у бюджетних установах, державних службах або громадських організаціях. Оскільки всі етапи

моделювання та аналізу виконуються у середовищі Jupyter Notebook, система є прозорою та гнучкою до адаптації під інші регіони або типи динамічних процесів. За потреби вона може бути розширена на економічні, фінансові чи воєнно-логістичні сценарії — тобто будь-які системи з часовою залежністю, які піддаються спостереженню.

Таким чином, прогнозуюча система, реалізована в межах кваліфікаційної роботи, є економічно доцільною. Вона дає змогу мінімізувати ризики, зменшити прямі й непрямі витрати у критичних ситуаціях, а також підвищити ефективність управлінських рішень на основі аналітичної моделі, що є адаптивною, масштабованою й придатною для автоматизації.

ВИСНОВКИ

У процесі виконання кваліфікаційної магістерської роботи на тему «Прогнозування динаміки складних процесів генетичними методами» було вирішено наукову та прикладну задачу побудови інтелектуальної системи, здатної здійснювати короткострокове прогнозування за часовими рядами з високим рівнем нелінійності, шуму та нестационарності. Предметне середовище дослідження — епідеміологічні дані про динаміку захворюваності на прикладі COVID-19 в Україні — було обрано з огляду на його практичну важливість, доступність статистичних даних і складність моделювання таких процесів традиційними методами.

На першому етапі дослідження було здійснено системний аналіз сучасних підходів до прогнозування динамічних процесів. Було виявлено, що класичні статистичні методи (ARIMA, експоненціальне згладжування тощо) є недостатньо ефективними в умовах змінної структури даних. Методи машинного навчання (особливо глибокого) мають потенціал, однак часто страждають від проблеми локальних мінімумів у процесі навчання. У зв'язку з цим було обґрунтовано доцільність використання генетичних алгоритмів (ГА) як засобу глобальної оптимізації початкових параметрів моделі.

Було реалізовано трьохшарову нейронну мережу прямого поширення (MLP) із 7 входами та 7 виходами, де кожен елемент вхідного та вихідного векторів відповідав дню у тижневій послідовності. Початкове налаштування ваг здійснювалося за допомогою генетичного алгоритму з чисельністю популяції 30 та 60 поколіннями. Після цього модель уточнювалася із застосуванням методу градієнтного спуску (SGD) на 200 епохах. Було розроблено власні функції для реалізації операцій скрещування, мутації, селекції та фітнес-оцінювання на основі середньоквадратичної похибки.

Особливу увагу було приділено формуванню навчальної вибірки: використовувався ковзний вікно-метод (7 на вході — 7 на виході), попередньо виконувалося масштабування даних у діапазон $[0,1]$ за допомогою `MinMaxScaler`.

Реалізація системи здійснювалася мовою програмування Python у середовищі Jupyter Notebook із використанням бібліотек pandas, numpy, torch, matplotlib.

У процесі тестування було побудовано графіки збіжності функції втрат, що продемонстрували стабільне зменшення похибки та відсутність перенавчання. Додатково було здійснено тестовий прогноз з середини датасету, результати якого візуалізувалися та порівнювалися з реальними значеннями. Отримані графіки підтвердили, що модель здатна повторювати загальні тенденції епідемічної динаміки, адаптуватися до нетипових змін і давати прогноз з відносною похибкою, меншою за 5–7% у більшості випадків.

Визначено, що найефективніше гібридне поєднання полягає у використанні ГА для глобального пошуку (початкове позиціонування в просторі параметрів) і локального уточнення через градієнтний метод. Це дозволило покращити стабільність та точність навчання моделі, уникнувши залежності від випадкової ініціалізації, яка є характерною проблемою для нейронних мереж.

Практична цінність роботи полягає у створенні прототипу прогнозної системи, що може бути використана в медичних установах, аналітичних службах або як частина інформаційно-аналітичних платформ у сфері громадського здоров'я. Також система є легко адаптованою до інших типів динамічних процесів — фінансових, соціальних, логістичних — що підтверджує універсальність обраного підходу. Результати моделювання можуть бути застосовані для оперативного виявлення загроз, планування ресурсів і підтримки прийняття рішень.

Апробація результатів відбулася під час участі у конференції «Сучасні інформаційні технології та системи в управлінні», де тема роботи була презентована у вигляді тез у співавторстві з Миронцовим М.Л. та Помазун О.М. Це дозволило не лише перевірити наукову новизну підходу, а й отримати конструктивний зворотний зв'язок для подальшого вдосконалення.

Таким чином, усі поставлені у роботі завдання було реалізовано. Було розроблено, протестовано й обґрунтовано метод побудови прогнозної моделі з використанням гібридного підходу на основі генетичних алгоритмів і нейронних

мереж. Отримані результати є придатними як для практичного впровадження, так і для подальших наукових досліджень у сфері прогнозування складних динамічних процесів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chen B., Han S., Liu X., Li Z., Chen T., Ji M. Prediction of an epidemic spread based on the adaptive genetic algorithm // *Frontiers in Physics*. 2023. Vol. 11. DOI: 10.3389/fphy.2023.1195087.
2. Akkaya A., Közkurt C. A Bibliometric Analysis of a Genetic Algorithm for Supply Chain Agility // *Mathematics*. 2024. Vol. 12, Issue 8. DOI: 10.3390/math12081199.
3. Al-Douri Y.K., Hamodi H., Lundberg J. Time Series Forecasting Using a Two-Level Multi-Objective Genetic Algorithm: A Case Study of Maintenance Cost Data for Tunnel Fans // *Algorithms*. 2018. Vol. 11, Issue 8. DOI: 10.3390/a11080123.
4. Al-Douri Y.K., Al-Chalabi H., Lundberg J. Time Series Forecasting Using Genetic Algorithm: A Case Study of Maintenance Cost Data for Tunnel Fans // *Proceedings of the Twelfth International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2018)*. 2018. P. 20–25.
5. Hammam R., Solyman A.A.A. ARIMA Model Estimation Based on Genetic Algorithm for COVID-19 Mortality Rate Forecasting // *Journal of Bioelectronics and Biomedical Engineering*. 2023. Vol. 5, Issue 1. P. 45–52.
6. Kumar S., Singh R. Autoregressive Integrated Moving Average (ARIMA) Model with Genetic Algorithm to Forecast the Chilli and Turmeric Productions in India // *Journal of Scientific Research and Reports*. 2023. Vol. 29, Issue 4. P. 123–130.
7. Boyko N.I., Blazhevskyy S.H. Методика визначення структури моделі оптимальної складності (порівняння генетичного алгоритму та ГМАР) // *Вісник Хмельницького національного університету*. 2022. № 2(307). С. 7–13.
8. Власенко Л.А. Дослідження генетичних алгоритмів та їх застосування у сфері алгоритмічного трейдингу // *Кваліфікаційна робота. Харківський національний університет радіоелектроніки*, 2021. 54 с.
9. Qiu Z., Zhang M., Wang Y., Li J., Li Y. Application of genetic algorithm combined with improved SEIR model in predicting the epidemic trend of COVID-19

(China) // *Scientific Reports*. 2022. Vol. 12. Article number: 8910. DOI: 10.1038/s41598-022-12958-z.

10. Pernin C.G., Moore M., Nixon M., McEver J., Cecchine G. Allocation of Forces, Fires, and Effects Using Genetic Algorithms // RAND Corporation Technical Report TR-423. 2008. 78 p.

11. Compare M., Zio E., Fumagalli L. Genetic Algorithms for Condition-Based Maintenance Optimization under Uncertainty // *European Journal of Operational Research*. 2019. Vol. 277, Issue 2. P. 508–522. DOI: 10.1016/j.ejor.2019.02.020.

12. Lin J., Zhang Y., Wang S., Liu Y. Time Series Forecasting by Adaptive Differential Evolution Algorithm Optimizing Neural Network Parameters // *Expert Systems with Applications*. 2015. Vol. 42, Issue 3. P. 855–863. DOI: 10.1016/j.eswa.2014.08.046.

13. Akkaya A., Közkurt C. Genetic Algorithm Research: A Bibliometric Analysis (1986–2024) // *Pioneer and Innovative Studies in Engineering*. 2024. P. 309–324.

14. Akkaya A., Közkurt C. A Bibliometric Analysis of a Genetic Algorithm for Supply Chain Agility // *Mathematics*. 2024. Vol. 12, Issue 8. P. 1199. DOI: 10.3390/math12081199.

15. Djunaidy A., Handoko L.T. A Review of the Applications of Genetic Algorithms to Forecasting Prices of Commodities. *Economies*. 2021. Vol. 9(1). Article 6. DOI: <https://www.mdpi.com/2227-7099/9/1/6>

16. Чопорова С.О., Ліснюк А.В. Використання генетичного алгоритму для оптимізації параметрів нейронної мережі при прогнозуванні напружено-деформованого стану квадратної пластинки. *Прикладні питання математичного моделювання*. 2022. №1(19). С. 66–71.

17. Пирогова Я.В., Ємеліаненко Т.Г. Розробка інформаційної технології короткострокового прогнозування нечітких часових рядів. *Актуальні проблеми автоматизації та інформаційних технологій*. 2021. №1(1). С. 34–41.

18. Khalaf R., Ghandour M., Rahal R., Rahme S. Hybrid Genetic Algorithms for Forecasting Power Systems State. *IEEE Conference Proceedings*. 2013.

19. Tran D.N., Xue B., Zhang M. DyForGP: A Genetic Programming Based Framework for Time Series Prediction. University of Adelaide Research Papers. 2017.
20. DataRobot Inc. DataRobot 7.1 Introduces Enhancements to Take AI Projects to the Next Level. Business Wire. 2021(дата звернення: 01.05.2025).
21. Saini H. TPOT AutoML: An Overview. GeeksforGeeks. 2023. URL: <https://www.geeksforgeeks.org/tpot-automl/>(дата звернення: 01.05.2025)
22. Robertson A. Symbolic Regression via Genetic Programming. GitHub Project. 2022. URL: <https://github.com/robertson809/symbolic-regression-genetic-programming> (дата звернення: 01.05.2025)
23. Lunatech Labs. The NEAT Algorithm: Evolving Neural Networks. Lunatech Blog. 2024. URL: <https://blog.lunatech.com/posts/2024-02-29-the-neat-algorithm-evolving-neural-network-topologies> (дата звернення: 01.05.2025)
24. NEAT-Python Documentation. NEAT Overview. NEAT-Python 0.92. URL: https://neat-python.readthedocs.io/en/latest/neat_overview.html (дата звернення: 01.05.2025)
25. Wood T. et al. Forecaster: A Genetic Algorithm-Based Optimization for Distributed Stream Processing. Proceedings of ICAC 2019. URL: <https://faculty.cs.gwu.edu/timwood/papers/19-ICAC-storm.pdf> (дата звернення: 01.05.2025)
26. Kaur H., Bansal A. Genetic Programming Based Approach Towards Understanding the System Dynamics. Procedia Computer Science. 2016. Vol. 89. P. 247–254.
27. Pytorch Documentation. Pytorch Documentation. URL: <https://pytorch.org/docs/stable/index.html> (дата звернення: 01.05.2025).
28. Pandas documentation. Pandas. 20.09.2024. URL: <https://pandas.pydata.org/docs/> (дата звернення: 01.05.2025).
29. NumPy Documentation. NumPy. 06.02.2020. URL: <https://numpy.org/doc/> (дата звернення: 01.05.2025).
30. Hunter J. D. H. Matplotlib: A 2D Graphics Environment. ResearchGate. 01.06.2007. URL:

https://www.researchgate.net/publication/3422921_Matplotlib_A_2D_Graphics_Environment (дата звернення: 01.05.2025). Pytorch Documentation. Pytorch Documentation. URL: <https://pytorch.org/docs/stable/index.html> (дата звернення: 01.05.2025).

31. Paszke A. Et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Pytorch. 07.05.2025. URL: <https://pytorch.org/> (дата звернення: 01.05.2025).

ДОДАТКИ

Додаток А – Код для прогнозування

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import torch.optim as optim
from sklearn.preprocessing import MinMaxScaler

file_path =
r'D:\ПТЕ\ДипломМагістр\Ukraine_covid_data_actualdate_Total_cases_regions.csv'
df = pd.read_csv(file_path, sep=';')

data = df['Ukraine'].values.reshape(-1, 1)

scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)

X, y = [], []
for i in range(len(data_scaled) - 14):
    X.append(data_scaled[i:i+7].flatten())
    y.append(data_scaled[i+7:i+14].flatten())

X = np.array(X, dtype=np.float32)
y = np.array(y, dtype=np.float32)

X_tensor = torch.tensor(X)
y_tensor = torch.tensor(y)

class NeuralNet(nn.Module):
    def __init__(self):
        super(NeuralNet, self).__init__()
        self.hidden = nn.Linear(7, 10)
        self.output = nn.Linear(10, 7)
        self.sigmoid = nn.Sigmoid()
```

```

def forward(self, x):
    x = self.sigmoid(self.hidden(x))
    x = self.sigmoid(self.output(x))
    return x

model = NeuralNet()

criterion = nn.MSELoss()
optimizer = optim.SGD(model.parameters(), lr=0.5)

losses = []
epochs = 200
for epoch in range(epochs):
    optimizer.zero_grad()
    outputs = model(X_tensor)
    loss = criterion(outputs, y_tensor)
    loss.backward()
    optimizer.step()
    losses.append(loss.item())

plt.plot(losses)
plt.title('Залежність похибки від кількості епох')
plt.xlabel('Епоха')
plt.ylabel('Середньоквадратична похибка')
plt.grid(True)
plt.show()

last_input = torch.tensor(data_scaled[-7:].reshape(1, -1), dtype=torch.float32)
prediction_scaled = model(last_input).detach().numpy()
prediction = scaler.inverse_transform(prediction_scaled.reshape(-1, 1))
print('Прогноз на наступні 7 днів:', prediction.flatten().round().astype(int))

```

Летич А.А., студент,
Миронцов М.Л., д. ф.-м. н.,
Помазун О.М., к.е.н.

*Київський національний економічний
університет імені Вадима Гетьмана*

letych.artem@kneu.ua

myrontsov@ukr.net

oksp@kneu.edu.ua

ПРОГНОЗУВАННЯ ДИНАМІКИ СКЛАДНИХ ПРОЦЕСІВ ГЕНЕТИЧНИМИ МЕТОДАМИ

Прогнозування динаміки складних процесів генетичними методами є одним із найперспективніших напрямів сучасної науки. Зі зростанням обсягів даних та необхідністю швидкої обробки інформації традиційні аналітичні моделі втрачають ефективність, оскільки не враховують нелінійні взаємозв'язки між факторами. Генетичні алгоритми дозволяють подолати ці обмеження, використовуючи принципи природного добору для пошуку оптимальних рішень у складних динамічних системах. Завдяки цьому вони знайшли застосування у прогнозуванні екологічних змін, фінансових криз, поширення інфекційних захворювань та багатьох інших процесів, які потребують моделювання невизначених сценаріїв розвитку.

Основним завданням прогнозування є своєчасне виявлення загроз та визначення сценаріїв їхнього можливого розвитку. У випадку біологічних, екологічних або соціальних процесів необхідно враховувати велику кількість змінних, включаючи поведінку індивідів, особливості навколишнього середовища, темпи розповсюдження певних явищ та їхню взаємодію з іншими процесами. З традиційної точки зору такі моделі ґрунтуються на диференціальних рівняннях, регресійному аналізі або стохастичних підходах, однак вони часто є статичними та не враховують можливих мутацій системи. Генетичні алгоритми натомість дозволяють змінювати параметри моделей у реальному часі, імітуючи природну еволюцію системи, що суттєво підвищує якість прогнозування.

Глобалізація та розвиток цифрових технологій призвели до того, що інформація поширюється майже миттєво. Це може бути використано як один із факторів раннього виявлення загроз, оскільки аналіз великих масивів текстових, аудіовізуальних та геолокаційних даних дозволяє знаходити приховані закономірності та визначати перші ознаки потенційної небезпеки. Використання обробки природної мови (NLP) у поєднанні з нейронними мережами дає змогу ідентифікувати ключові події та тренди, що випереджають фактичну появу явища у фізичному просторі.

Важливою особливістю прогнозування складних процесів є необхідність постійного оновлення даних і модифікації моделей. Наприклад, при аналізі динаміки поширення вірусних інфекцій виявлено, що кількість зареєстрованих випадків може змінюватися залежно від днів тижня, сезонних факторів чи адміністративних рішень. Тому прогнозування на основі щоденних даних може бути менш ефективним порівняно з використанням динаміки за більші часові періоди, такі як тиждень або місяць. Це дозволяє мінімізувати вплив короткострокових флуктуацій і отримати більш стабільні тренди.

Використання генетичних алгоритмів у прогнозуванні складних процесів передбачає не лише апроксимацію існуючих даних, а й активне навчання моделі з кожним новим оновленням інформації. Це відрізняється від традиційних підходів, які часто використовують зафіксовані параметри та не адаптуються до змінних умов. Інтеграція національних та міжнародних джерел даних дозволяє отримувати більш повну картину та забезпечувати гнучке реагування на нові виклики. Водночас існує проблема надійності інформації, оскільки соціальні мережі та медіа-простір можуть містити велику кількість фейкових новин або навмисно маніпулятивних

повідомлень. Саме тому особливу увагу необхідно приділяти верифікації даних та створенню алгоритмів, які здатні відрізнити реальні загрози від інформаційного шуму.

Сучасні підходи до прогнозування включають багаторівневий аналіз даних, що поєднує різні методи штучного інтелекту, математичного моделювання та кібернетики. Важливою складовою є автоматизоване поєднання різних типів інформації – текстової, візуальної, сенсорної, геопросторової – в єдиний аналітичний простір. Це дозволяє створити ефективні системи моніторингу та прогнозування, які працюють у режимі реального часу.

Результати таких розробок можуть бути використані в різних сферах – від охорони здоров'я та екології до національної безпеки та фінансових ринків. Наприклад, автоматизовані системи моніторингу можуть допомагати виявляти потенційні економічні кризи шляхом аналізу поведінки інвесторів і динаміки фондових ринків. У сфері екології прогнозування на основі генетичних алгоритмів дозволяє моделювати зміни клімату та вплив антропогенних факторів на довкілля.

У військовій сфері прогнозування загроз набуває особливого значення. Зокрема, у зонах бойових дій або підвищеної міграції населення ризик епідемій суттєво зростає через погіршення санітарних умов і перевантаженість медичних установ. У таких умовах необхідно розробляти швидкодіючі системи, здатні аналізувати потоки інформації та прогнозувати можливі сценарії розвитку подій.

Отже, використання генетичних методів у прогнозуванні складних процесів є перспективним напрямом, що дозволяє підвищити точність моделей, адаптувати їх до мінливих умов та знаходити оптимальні рішення для запобігання кризовим ситуаціям. Реалізація таких підходів у поєднанні з сучасними технологіями штучного інтелекту сприятиме розвитку ефективних систем управління ризиками та забезпеченню стійкості суспільства до нових викликів.

Список використаних джерел

1. Genetic algorithm. *Wikipedia*. 31.03.2025. URL: https://en.wikipedia.org/wiki/Genetic_algorithm (дата звернення: 31.03.2025).
2. Sourabh K., Sumit S. C., Vijay K. A review on genetic algorithm: past, present, and future. *Springer*. 31.10.2020. URL: <https://link.springer.com/article/10.1007/s11042-020-10139-6> (дата звернення: 31.03.2025).
3. Yuhai T. Learning dynamical systems from data. *PNAS*. 24.07.2024. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.2311808121> (дата звернення: 31.03.2025).

Науковий керівник: Миронцов М.Л., доктор фізико-математичних наук.



Звіт подібності

метадані

Назва організації

Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок

ПРОГНОЗУВАННЯ ДИНАМІКИ СКЛАДНИХ ПРОЦЕСІВ ГЕНЕТИЧНИМИ МЕТОДАМИ

Автор

Науковий керівник / Експерт

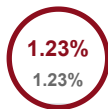
Летич Артем Анатолійович Миронцов М.Л.

підрозділ

кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1

25

Довжина фрази для коефіцієнта подібності 2



КП 2

16813

Кількість слів



КЦ

134555

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		5
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		14

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ir.kneu.edu.ua/bitstreams/e9391672-b25d-4bf5-a6fa-9a0664aee9d8/download	34 0.20 %
2	https://ir.kneu.edu.ua/bitstreams/e9391672-b25d-4bf5-a6fa-9a0664aee9d8/download	25 0.15 %
3	https://ir.kneu.edu.ua/bitstreams/e9391672-b25d-4bf5-a6fa-9a0664aee9d8/download	22 0.13 %
4	Статья Али Нуржан 11/25/2024 Taraz University named after M. Kh. Dulati (каф. Информационные системы)	18 0.11 %

5	https://ru.essays.club/%D0%A2%D0%BE%D1%87%D0%BD%D1%8B%D0%B5-%D0%BD%D0%B0%D1%83%D0%BA%D0%B8/%D0%9A%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8-%D0%B8-%D1%81%D0%B2%D1%8F%D0%B7%D1%8C/%D0%97%D0%B0%D1%85%D0%B8%D1%81%D1%82-%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97-%D0%B7-%D0%BE%D0%B1%D0%BC%D0%B5%D0%B6%D0%B5%D0%BD%D0%B8%D0%BC-470343.html	15 0.09 %
6	магістерська Менделюк 12/19/2024 Ivano-Frankivsk National Technical University of Oil and Gas (Каф. КСМ)	11 0.07 %
7	https://ir.kneu.edu.ua/bitstreams/e9391672-b25d-4bf5-a6fa-9a0664aee9d8/download	11 0.07 %
8	https://ir.kneu.edu.ua/bitstreams/e9391672-b25d-4bf5-a6fa-9a0664aee9d8/download	11 0.07 %
9	https://ir.kneu.edu.ua/bitstreams/e9391672-b25d-4bf5-a6fa-9a0664aee9d8/download	10 0.06 %
10	https://ir.kneu.edu.ua/bitstreams/e9391672-b25d-4bf5-a6fa-9a0664aee9d8/download	9 0.05 %

з бази даних RefBooks (0.00 %)



ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-----------	--

з домашньої бази даних (0.08 %)



ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Формування та реалізація публічної політики сталого розвитку територій в Україні 5/7/2020 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра національної економіки та публічного управління)	13 (2) 0.08 %

з програми обміну базами даних (0.21 %)



ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Статья Али Нуржан 11/25/2024 Taraz University named after M. Kh. Dulati (каф. Информационные системы)	18 (1) 0.11 %
2	магістерська Менделюк 12/19/2024 Ivano-Frankivsk National Technical University of Oil and Gas (Каф. КСМ)	11 (1) 0.07 %
3	Курсова робота Андріюк Владислав.doc 12/19/2023 Lesya Ukrainka Volyn National University (Кафедра практики англійської мови)	7 (1) 0.04 %

з Інтернету (0.94 %)



ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ir.kneu.edu.ua/bitstreams/e9391672-b25d-4bf5-a6fa-9a0664aee9d8/download	138 (10) 0.82 %
2	https://ru.essays.club/%D0%A2%D0%BE%D1%87%D0%BD%D1%8B%D0%B5-%D0%BD%D0%B0%D1%83%D0%BA%D0%B8/%D0%9A%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8-%D0%B8-%D1%81%D0%B2%D1%8F%D0%B7%D1%8C/%D0%97%D0%B0%D1%85%D0%B8%D1%81%D1%82-%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97-%D0%B7-%D0%BE%D0%B1%D0%BC%D0%B5%D0%B6%D0%B5%D0%BD%D0%B8%D0%BC-470343.html	20 (2) 0.12 %