Міністерство освіти і науки України
Донецький національний університет імені Василя Стуса, м. Вінниця
Київський національний університет будівництва і архітектури
Черкаський державний технологічний університет
Київський національний торговельно-економічний університет
Львівський національний університет імені Івана Франка
Тернопільський національний технічний університет імені Івана Пулюя
Громадська організація «Освітня фундація продуктового ІТ»

Матеріали

V ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ «КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ОБРОБКИ ДАНИХ»

6 грудня 2024 року

Матеріали надруковані в авторській редакції.
Достовірність поданої інформації лежить на авторах публікацій

Вінниця
2024

УДК 004.9:[001.8:378.091.5]’06(043.2)

Рекомендовано до друку вченою радою факультету інформаційних і прикладних технологій Донецького національного університету імені Василя Стуса (протокол № 2(6) від 19.12.2024)

Комп’ютерні технології обробки даних: матеріали V Всеукраїнської науково-практичної конференції (м. Вінниця, 06 грудня 2024 р.). Вінниця: ДонНУ імені Василя Стуса, 2024. 168 с.

Збірник містить матеріали V Всеукраїнської науково-практичної конференції «Комп’ютерні технології обробки даних». Тематика конференції охоплює актуальні питання досліджень у сфері інформаційних технологій, зокрема методів обробки та аналізу даних, штучного інтелекту, інтелектуальних систем та мереж, баз даних і знань, прикладних аспектів комп’ютерних технологій, кібербезпеки, інтернету речей та математичних методів моделювання. Матеріали учасників конференції адресовано фахівцям та усім, хто цікавиться сучасним станом вивчення прикладних аспектів сучасних міждисциплінарних досліджень.

ISSN (print): 2788-4465 **УДК 004.9:[001.8:378.091.5]’06(043.2)**

ЗМІСТ

СЕКЦІЯ 1. МЕТОДИ ОБРОБКИ ТА АНАЛІЗУ ДАНИХ 9

Байраківська В. В., Сеник І. О.
ГРАФИ ТА ЇХ ВИКОРИСТАННЯ В АЛГОРИТМІЗАЦІЇ 10

Загаєцька А. М., Комаров В. Ф.
ПЕРСПЕКТИВИ РОЗВИТКУ КОМП'ЮТЕРНО-МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ 12

Іваненко А. В., Січко Т. В.
РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРОГНОЗУВАННЯ СПОЖИВЧОГО ПОПИТУ МЕТОДАМИ ШТУЧНОГО ІНТЕЛЕКТУ 14

Канюка Р. Ю., Ротштейн О. П.
МАТЕМАТИЧНА ОБРОБКА ОТРИМАНИХ ДАНИХ УЛЬТРАЗВУКОВОГО ДАТЧИКА 17

Кизь Н. А., Луценко А. В.
МЕТОДИ ОБРОБКИ ТА АНАЛІЗУ ДАНИХ 19

Курбангалієв Д. М., Поремський Ю. В.
АЛГОРИТМИ МАШИННОГО НАВЧАННЯ 21

Лавренюк Б. В., Потапова Н. А.
ВИКОРИСТАННЯ АНАЛІЗУ ДАНИХ ДЛЯ ПРОГНОЗУВАННЯ ТА ВИЯВЛЕННЯ ТРЕНДІВ У СВІТІ КІНЕМАТОГРАФІЇ 24

Лаптєва М. А., Хмелівський Ю. С.
ВИЯВЛЕННЯ УПЕРЕДЖЕНЬ У ШІ ЗА ДОМОПОГОЮ СТАТИСТИЧНОГО НАВЧАННЯ 26

Левченко М. Р., Хмелівський Ю. С.
АНАЛІЗ ВИЖИВАННЯ ПАСАЖИРІВ НА БОРТУ TITANIC ЗА ДОПОМОГОЮ DATASET У МОВІ R 29

Ліхоткіна В. І., Потапова Н. А.
СУТНІСТЬ ТА СКЛАДНИКИ РЕКУРСИВНИХ АЛГОРИТМІВ У ПРОГРАМУВАННІ 32

Петров Д. Ю., Поремський Ю. В.
ХЕШ-ТАБЛИЦІ: КОНЦЕПЦІЇ, РЕАЛІЗАЦІЯ ТА ПРИКЛАДИ ВИКОРИСТАННЯ У ПРОГРАМУВАННІ 34

Радченко Р. П., Сеник І. О.
ГРАФИ ТА АЛГОРИТМИ ПОШУКУ: ТЕОРІЯ, ЗАСТОСУВАННЯ ТА ОПТИМІЗАЦІЯ 36

Ратушний О. С., Сеник І. О.
ОСОБЛИВОСТІ ВИКОРИСТАННЯ СТРУКТУРИ ДАНИХ «ГРАФ»
У РОЗРОБЦІ ВІДЕОІГОР 38

Родюк К. О., Луценко А. В.
ЗАСТОСУВАННЯ МАТЕМАТИКИ У ГЛИБИННОМУ НАВЧАННІ 40

Слободяник Р. Р., Поремський Ю. В.
ОСОБЛИВОСТІ ВИКОРИСТАННЯ АЛГОРИТМІВ У КІБЕРБЕЗПЕЦІ 42

Труханська В. О., Ніколюк П. К.
ОБРОБКА СИГНАЛІВ ДЛЯ ВИЯВЛЕННЯ ПОВІТРЯНИХ ЦІЛЕЙ 44

Хріпко О. І., Фриз І. В.
РОЗВ'ЯЗУВАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ
НА МОВІ PYTHON 47

Чемес В. С., Хмелівський Ю. С.
ЗАСТОСУВАННЯ ЛІНІЙНОЇ РЕГРЕСІЇ ДЛЯ ОЦІНКИ ЦІНИ
НЕРУХОМОСТІ ЗА РІЗНИМИ ПАРАМЕТРАМИ 49

Черніхевич О. В., Поремський Ю. В.
АЛГОРИТМИ ДИНАМІЧНОГО ПРОГРАМУВАННЯ В ЗАДАЧАХ
ОПТИМАЛЬНОГО ВИБОРУ 52

СЕКЦІЯ 2. ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ТА МЕРЕЖІ 54

Зигарь Д. Д., Ніколюк П. К.
РОЗРОБЛЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЇ
ЦІЛЕЙ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ 55

Бурківський О. С., Ніколюк П. К.
ОПТИМІЗАЦІЯ МАРШРУТІВ ПОСТАЧАННЯ ВІЙСЬКОВОЇ ТЕХНІКИ
ЗА ДОПОМОГОЮ A*-АЛГОРИТМУ 58

Дорофєєв Є. О., Січко Т. В.
РОЛЬ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ (BIG DATA) У СУЧАСНИХ
ІНОРМАЦІЙНИХ СИСТЕМАХ 61

Поліщук В. С., Ніколюк П. К.
РОЗРОБКА МОДЕЛІ АКУСТИЧНОГО ЛОКАТОРА ДЛЯ ВИЯВЛЕННЯ
ДРОНІВ СУПРОТИВНИКА НА НИЗЬКИХ ВИСОТАХ 63

Семен О. Д., Ніколюк П. К.
ІНТЕГРАЦІЯ ШТУЧНОГО ІНТЕЛЕКТУ У ТРАНСПОРТНІ СИСТЕМИ
ДЛЯ ОПТИМІЗАЦІЇ МІСЬКОЇ ІНФРАСТРУКТУРИ 66

Слободянюк С. С., Якубич К. О.
ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В СИСТЕМНОМУ
МОДЕЛЮВАННІ 69

Щербина Д. С., Ніколюк П. К.
ЗАСТОСУВАННЯ МОДЕЛІ YOLO ДЛЯ РОЗПІЗНАВАННЯ ВОРОЖОЇ ВІЙСЬКОВОЇ ТЕХНІКИ 71

СЕКЦІЯ 3. ЕКСПЕРТНІ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ ТА СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ 73

Чуприна І. А., Антонов Ю. С.
ОСОБЛИВОСТІ РОЗРОБКИ СИСТЕМИ АВТОМАТИЧНОГО СТВОРЕННЯ ТА ВАЛІДАЦІЇ ТЕСТОВИХ ПИТАНЬ НА ОСНОВІ ВЗАЄМОДІЇ З CHATGPT 74

Sapozhnikova V., Khmelivskyi Y.
ADVERTISING STRATEGY: A GAME THEORY APPROACH 76

Козачок А. О., Хмелівський Ю. С.
ЗАСТОСУВАННЯ СИСТЕМИ РЕКОМЕНДАЦІЙ У ОНЛАЙН-МАГАЗИНІ 79

Мигун Р. С., Потапова Н. А.
ЗАСТОСУВАННЯ АЛГОРИТМУ ПОБУДОВАНОЇ ЦИФРОВОЇ ПІДПИСКИ З ВИКОРИСТАННЯМ ЕЛІПТИЧНИХ КРИВИХ (ECDSA) У БЛОКЧЕЙН-ТЕХНОЛОГІЯХ 81

Перепелиця А. С., Бабаков Р. М.
ІНФОРМАЦІЙНА ПЛАТФОРМА ДЛЯ АНАЛІЗУ НАВЧАННЯ ТА КАР'ЄРНИХ МОЖЛИВОСТЕЙ 84

Плець В. В., Потапова Н. А.
ВИКОРИСТАННЯ ХЕШ-ФУНКЦІЙ І ХЕШ-ТАБЛИЦЬ У КІБЕРБЕЗПЕЦІ 87

Русавський О. О., Ротштейн О. П.
ІДЕНТИФІКАЦІЯ НЕЛІНІЙНИХ ЗАЛЕЖНОСТЕЙ НЕЧІТКИМИ КОГНІТИВНИМИ КАРТАМИ 90

Семенюк А. М., Якубич К. О.
СИСТЕМА ЕКСПЕРТНОЇ ОЦІНКИ НА БАЗІ FUZZY LOGIC 92

Сидорчук Р. В., Потапова Н. А.
ГЕКСАГОНАЛЬНА АРХІТЕКТУРА В РОЗРОБЦІ УНІВЕРСАЛЬНИХ ВЕБДОДАТКІВ 95

Яценко В. В., Сеник І. О.
ЗАСТОСУВАННЯ МЕТОДІВ ОПТИМІЗАЦІЇ ДЛЯ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ ІЗ ВЕЛИКИМ ОБСЯГОМ ПАРАМЕТРІВ 97

СЕКЦІЯ 4. БАЗИ ДАНИХ І ЗНАНЬ. BIG DATA.....100

Ілик В. В., Якубич К. О.
ПРОГРЕСИВНІ ВЕБДОДАТКИ (PWA): ІННОВАЦІЙНІ МОЖЛИВОСТІ ДЛЯ ІНТЕРАКТИВНИХ ПЛАТФОРМ І ЇХ РОЛЬ У ТРАНСФОРМАЦІЇ КОРИСТУВАЦЬКОГО ДОСВІДУ.....101

Рудь О. С., Зелінська О. В.
BIG DATA У ТРАНСПОРТНИХ МЕРЕЖАХ 104

СЕКЦІЯ 5. ПРИКЛАДНІ АСПЕКТИ ОБРОБКИ ДАНИХ В ІНФОРМАЦІЙНИХ СИСТЕМАХ.....106

Антонюк А. О., Луценко А. В.
ЗАСТОСУВАННЯ МАТЕМАТИЧНОЇ СТАТИСТИКИ В АНАЛІЗІ ДАНИХ.....107

Баліцький В. В., Римар П. В.
ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ ПРОЦЕСАМИ ВЕТЕРИНАРНОЇ КЛІНІКИ.....109

Зелінський О. О., Потапова Н. А.
ОСОБЛИВОСТІ ВИКОРИСТАННЯ ГІБРИДНИХ МЕТОДОЛОГІЙ УПРАВЛІННЯ ПРОЄКТАМИ 110

Ковальчук А. А., Римар П. В.
ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ MAPLE ДЛЯ ПОБУДОВИ КРИВИХ ТА ПОВЕРХОНЬ ТРЕТЬОГО ПОРЯДКУ 113

Костенко Р. О., Зелінська О. В.
РОЛЬ ПРОЄКТНОГО МЕНЕДЖЕРА В ЦИФРОВІЙ ТРАНСФОРМАЦІЇ МІЖНАРОДНИХ ОРГАНІЗАЦІЙ 116

Круцюк Д. О., Римар П. В.
ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ MAPLE ДЛЯ РОЗВ’ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ 118

Курдупов О. Л., Антонов Ю. С.
РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕДИЧНОГО ЗАКЛАДУ НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН.....121

Куцмай В. Я., Зелінська О. В.
УПРАВЛІННЯ МУЛЬТИНАЦІОНАЛЬНИМИ КОМАНДАМИ В УМОВАХ ЦИФРОВОЇ ЕКОНОМІКИ ТА НЕСТАБІЛЬНОСТІ 123

Мазур А. В., Римар П. В.
ВИКОРИСТАННЯ MS EXCEL ДЛЯ РОЗВ’ЯЗАННЯ СТАТИСТИЧНИХ ЗАДАЧ.....125

Мельник В. Р., Якубич К. О.
МОДЕЛЮВАННЯ ОПТИМАЛЬНОГО ВІДНОВЛЕННЯ
ЕНЕРГОСИСТЕМ ПІСЛЯ ПОШКОДЖЕНЬ 128

Михайляк М. О., Комаров В. Ф.
МОДЕЛЮВАННЯ ФІЗИЧНИХ СИСТЕМ ЗАСОБАМИ MAPLE
ТА SIMULINK 130

Мишківська Я. В., Січко Т. В.
АНАЛІЗ ПРОДУКТИВНОСТІ ХУКІВ У ПОРІВНЯННІ З КЛАСОВИМИ
КОМПОНЕНТАМИ 132

Морозюк А. А., Зелінська О. В.
ІНТЕГРАЦІЯ ПРИНЦИПІВ ПРОДУКТОВОЇ АНАЛІТИКИ
У ДИЗАЙНІ ІНФОРМАЦІЙНИХ СИСТЕМ 134

Морозюк А. А., Комаров В. Ф.
КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЯК ІНСТРУМЕНТ БОРОТЬБИ
ІЗ ТРАНСПОРТНИМИ ЗАТОРАМИ 137

Наральник Б. Ю., Ніколюк П. К.
ОБРОБКА ДАНИХ У ХМАРНИХ СЕРЕДОВИЩАХ: ВИКЛИКИ
ТА ПЕРСПЕКТИВИ 139

Підруцький Д. А., Комаров В. Ф.
КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЕКОЛОГІЧНИХ СИСТЕМ,
ЩО ЗАЗНАЛИ ШКОДИ ПІД ЧАС ВІЙНИ 141

Синіцький В. Ю., Поремський Ю. В.
РЕКУРСИВНІ АЛГОРИТМИ В СУЧАСНОМУ ПРОГРАМУВАННІ 142

Тищенко О. М., Поремський Ю. В.
АЛГОРИТМІЗАЦІЯ В ПОСТАНОВЦІ І РОЗВ'ЯЗАННІ ВІЙСЬКОВИХ
ЗАВДАНЬ 144

Тітаренко Р. А., Комаров В. Ф.
ЗАСТОСУВАННЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ
ДЛЯ ОПТИМІЗАЦІЇ РОЗПОДІЛУ ЕЛЕКТРОЕНЕРГІЇ В МІСЬКИХ
МЕРЕЖАХ 146

Трохимчук О. М., Якубич К. О.
РОЛЬ АЛГОРИТМІЗАЦІЇ У РОЗВ'ЯЗАННІ ЗАДАЧ
ШТУЧНОГО ІНТЕЛЕКТУ 147

Чайковський П. А., Штовба С. Д.
ПОРІВНЯЛЬНИЙ АНАЛІЗ ТЕХНІК FEW-SHOT LEARNING
ДЛЯ ДОМЕННОЇ АДАПТАЦІЇ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ 149

Чемес В. С., Сеник І. О.
АНАЛІЗ І ЗАСТОСУВАННЯ МОДЕЛЕЙ ТЕОРІЇ ЧЕРГ
У СИСТЕМАХ ОБСЛУГОВУВАННЯ 151

Чулюк В. В., Якубич К. А.
СУТНІСТЬ ТА ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МЕТОДУ СОРТУВАННЯ ВСТАВКОЮ 155

Щербина Д. С., Потапова Н. А.
ЗАСТОСУВАННЯ АЛГОРИТМУ АЛЬФА-БЕТА ВІДСІКАННЯ ДЛЯ ОПТИМІЗАЦІЇ ШАХОВИХ РІШЕНЬ 157

Рудь О. С., Юстименко Є. А.
АЛГОРИТМИ ШИФРУВАННЯ ДЛЯ ЗАХИСТУ РАДІОЗВ'ЯЗКУ В УМОВАХ БОЙОВИХ ДІЙ 159

Ярошенко Б. М., Хмелівський Ю. С.
РОЛЬ АЛГОРИТМІВ У БЛОКЧЕЙН-ТЕХНОЛОГІЯХ 162

Кузьміна М. О., Якубич К. О.
РОЗРОБКА ПРОГРАМИ ДЛЯ КУХАРІВ НА PYTHON 163

Ілик В. В., Сеник І. О.
АРХІТЕКТУРА МІКРОСЕРВІСІВ ДЛЯ МАСШТАБНИХ КЛІЄНТ-СЕРВЕРНИХ СИСТЕМ 165

СЕКЦІЯ 1
МЕТОДИ ОБРОБКИ ТА АНАЛІЗУ ДАНИХ

УДК 004.021:004.42;519.17

Байраківська В. В., здобувач вищої освіти,
Сеник І. О., асистент кафедри
інформаційних технологій

ГРАФИ ТА ЇХ ВИКОРИСТАННЯ В АЛГОРИТМІЗАЦІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

Графи є математичним інструментом для моделювання та розв'язання задач у комп'ютерних науках, інженерії, економіці та інших галузях. Для роботи з елементами в графах використовують зв'язки між вузлами та ребра, що дає змогу ефективного застосування у розв'язанні задач, пов'язаних із пошуком шляхів, маршрутизацією, розумним розподілом ресурсів та іншим. З цією метою графи використовуються в різних індустріях – від соціальних мереж до вантажних перевезень та телекомунікаційних систем. Такі алгоритми розв'язання задач на графах діють і допомагають працювати з даними, представлені у вигляді графів.

Граф є структурою даних, яка визначає спосіб опису зв'язків між набором елементів. Граф складається з набору об'єктів, які називаються вузлами, причому певні пари цих об'єктів з'єднані зв'язками, які називаються ребрами. Говоримо, що два вузли є сусідніми, якщо вони з'єднані ребром. Розрізняють орієнтовний та неорієнтовний графи, мультиграф та псевдограф. У орієнтовного графа ребра називаються дугами [1].

Для ефективного представлення графів у пам'яті комп'ютера використовують два найбільш поширені способи: матриця суміжності та матриця інциденцій.

Матриця суміжності будується у вигляді таблиці, водночас, якщо в таблиці стоїть 1 – то дві вершини суміжні. В іншому випадку, якщо в таблиця стоїть 0 – то дві вершини між собою не суміжні.

Матриця інциденцій також будується у вигляді таблиці, водночас у таблиці зображено зв'язки між інцидентними елементами графу (дуга і вершина). Стовпці матриці відповідають дугам, рядки – вершинам. Якщо в таблиці стоїть 1 – то ця дуга є вихідною з цієї вершини. Якщо в таблиці стоїть –1 – то ця дуга є вхідною в цю вершину. В іншому випадку, якщо в таблиця стоїть 0 – то ця дуга не має ніякого зв'язку з цією вершиною.

Поняття графу дає змогу структурувати інформацію та алгоритмізувати цілу низку типових задач. Саме тому існує багато алгоритмів, які працюють з інформацією, що представлена у вигляді графів. Їх ще називають «алгоритми на графах». Зокрема, щоразу, коли ви відкриваєте файл на комп'ютері, останній запускає алгоритм пошуку в пам'яті адреси файла. Під час цього пошук виконується з використанням алгоритму на графах [2].

Розглянемо основні алгоритми на графах. **Пошук в ширину:** алгоритм, що систематично досліджує вершини графу на мінімальній відстані від початкової вершини, знаходить всі досяжні вершини, обчислює відстань до них і створює «дерево в ширину», що охоплює ці вершини. Алгоритм працює як на

орієнтованих, так і на неорієнтованих графах [3]. Використовується в соціальних мережах для виявлення всіх друзів на певній відстані від заданого користувача, у штучному інтелекті для пошукових алгоритмів та у маршрутизації комп'ютерних мереж [4].

Пошук в глибину: алгоритм, який намагається йти якомога глибше в графі від початкової вершини, досліджуючи вершини вздовж кожного шляху до кінця. Коли більше немає сусідніх невідвіданих вершин, алгоритм повертається назад до попередньої вершини і продовжує пошук, поки не відвідає всі вершини графу [3]. Використовується в задачах з обмеженнями, у штучному інтелекті для зберігання станів у процесі пошуку та у соціальних мережах для виявлення циклів [5].

Алгоритм Дейкстри: знаходить найкоротші шляхи від однієї початкової вершини в орієнтованому графі з невід'ємними вагами ребер. Застосовується у маршрутизації, геолокаційних сервісах та транспортних системах [3].

Алгоритм Беллмана–Форда знаходить найкоротші шляхи в графах, де можливі від'ємні ваги ребр і виявляє цикли з від'ємною вагою. Цей алгоритм моделює фінансові мережі, маршрутизацію та транспортну логістику [3].

Алгоритм Прима будує мінімальне остовне (кістякове) дерево, додаючи до дерева ребра з найменшою вагою. Пріоритетна черга використовується для швидкого вибору наступного ребра. Використовується для проєктування мереж, кластеризації даних та створення електронних схем [3].

Алгоритм Крускала будує мінімальне остовне дерево, додаючи найменш важкі ребра, які з'єднують окремі частини графу. Допомагає в оптимізації розташування кабелів, трубопроводів та кластеризації даних [3].

Метод Форда–Фалкерсона розв'язує задачу максимального потоку в графі, ітеративно збільшуючи потік. Початковий потік дорівнює нулю, і на кожному кроці знаходиться збільшувальний шлях у резервній мережі, щоб підвищити потік. Є важливим для оптимізації транспорту, ланцюгів постачання та телекомунікаційних мереж [3].

Алгоритми на графах мають широкий спектр застосувань у реальному житті. Вони використовуються в соціальних мережах, мережах передачі даних, логістиці, фінансах і багатьох інших галузях для оптимізації процесів, зменшення витрат і покращення ефективності. Ці алгоритми є основою для багатьох сучасних технологій, включаючи планування маршрутів, аналіз даних та управління мережами.

Список використаних джерел

1. Easley D., Kleinberg J. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Book, Cambridge University Press, 2010, 744 p.

2. Кузьменко І. М. Теорія графів: посібник. Київ: КПІ ім. Ігоря Сікорського. 2020. 71 с.

3. Introduction To Algorithms / T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein. The MIT Press Cambridge, Massachusetts London, England. 2009. 1313 p.

4. Sedgewick R., Wayne K. Algorithms. Book, Addison-Wesley Professional. 2011. 976 p.

5. Kleinberg J., Tardos É. Algorithm Design. Book, Addison-Wesley. 2006. 864 p.

УДК 004.94:519.6

Загаєцька А. М., здобувачка вищої освіти,
Комаров В. Ф., канд. техн. наук,
старший викладач кафедри прикладної
математики та кібербезпеки

ПЕРСПЕКТИВИ РОЗВИТКУ КОМП'ЮТЕРНО-МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Комп'ютерно-математичне моделювання (КММ) є важливою частиною сучасних наукових досліджень і технічних розробок. Воно дає змогу аналізувати складні процеси, оцінювати їх наслідки, та моделювати ситуації, що недоступні для прямого експерименту. Проте стрімкий розвиток технологій потребує вдосконалення методів і інструментів КММ. Розглянемо перспективні напрями розвитку: впровадження штучного інтелекту, застосування квантових обчислень і використання хмарних технологій [1].

Основний текст. Штучний інтелект (ШІ) активно інтегрується у наукову сферу, спрощуючи складні обчислення і підвищуючи точність моделей. Його впровадження дає змогу вирішувати завдання, які раніше були занадто складними або потребували значного часу.

Традиційно створення математичних моделей вимагало глибоких знань і великих витрат часу. Сучасні алгоритми ШІ, зокрема методи машинного навчання, дають змогу автоматизувати цей процес. ШІ має здатність працювати з великими та складними наборами даних. Наприклад, у сфері фінансів моделі, що використовують ШІ, аналізують багаторічні дані фондових ринків для прогнозування їх поведінки. Завдяки цьому інвестори можуть краще оцінювати ризики. Генеративні змагальні мережі (GAN) допомагають створювати високоточні симуляції, що знаходять застосування в медицині для віртуального тестування ліків або в астрономії для моделювання динаміки галактик. Методи машинного навчання, як-от нейронні мережі, дають змогу покращити параметри моделі без втручання людини. У промисловості це використовується для оптимізації виробничих процесів, зменшення енергоспоживання та підвищення продуктивності [2].

Квантові обчислення – це новий етап у розвитку комп'ютерних технологій, який пропонує унікальні можливості для складних математичних задач. Вони базовані на принципах квантової механіки, пропонують революційні рішення для КММ:

- квантові комп'ютери можуть виконувати паралельні обчислення на рівні, який неможливий для традиційних пристроїв. Наприклад, в обчислювальній хімії вони використовуються для моделювання складних молекул. Замість місяців роботи класичного комп'ютера квантовий комп'ютер може виконати обчислення за кілька годин;

- багатовимірні рівняння, які потребують значних обчислювальних ресурсів, можна розв'язувати швидше та точніше завдяки квантовим алгоритмам;
- симуляція поведінки молекул або матеріалів у хімії та фізиці може здійснюватися з безпрецедентною точністю.

Хмарні обчислення надають платформу для масштабування можливостей КММ. Хмарні платформи, як-от AWS, Google Cloud і Microsoft Azure, дають змогу виконувати обчислення на потужних серверах, не купуючи дороге обладнання. Це особливо актуально для університетів та науково-дослідних центрів. Хмарні сервіси допомагають користувачам з обмеженими обчислювальними ресурсами виконувати складні моделювання на потужних серверах. Використання хмарних сховищ спрощує доступ до моделей і результатів досліджень для команд з усього світу. Наприклад, у медицині це дає змогу спільно аналізувати генетичні дані та моделювати нові підходи до лікування. Хмарні технології допомагають орендувати ресурси лише на час обчислень, знижуючи витрати на інфраструктуру. Це робить складні моделювання доступними навіть для стартапів і малих підприємств [3].

Комп'ютерно-математичне моделювання (КММ) – це потужний інструмент, який трансформує різні сфери життя, сприяючи ефективності, інноваціям та точності в дослідженнях і розробках.

У **медичній сфері** комп'ютерно-математичне моделювання є ключовим інструментом для аналізу, прогнозування та створення інноваційних методів лікування:

1. Прогнозування захворювань – моделі на основі ШІ здатні передбачати розвиток складних захворювань, як-от рак, діабет або серцево-судинні порушення.

2. Симуляція ефективності ліків – перед клінічними випробуваннями ліки тестуються на моделях, які відтворюють роботу людських органів чи систем. Це значно скорочує витрати часу і ресурсів, одночасно підвищуючи безпеку для пацієнтів.

3. Система охорони здоров'я – моделі допомагають у прогнозуванні потреб у медичних ресурсах, як-от ліжко-місця, медичне обладнання чи вакцини, що було критично важливим під час пандемії COVID-19.

КММ робить вагомий внесок у **розвиток сталої енергетики та ефективного управління ресурсами.**

1. Оптимізація роботи електромереж – сучасні електромережі інтегрують різні джерела енергії – традиційні та відновлювані. КММ дає змогу аналізувати й моделювати навантаження на мережі, оптимізувати розподіл енергії та запобігати збоям у системі.

2. Розробка технологій для відновлюваної енергетики – у галузі сонячної та вітрової енергетики моделювання використовується для аналізу продуктивності станцій за різних погодних умов.

3. Управління ресурсами – моделі допомагають ефективніше використовувати традиційні джерела енергії, як-от нафта й газ, з мінімізацією екологічного впливу.

Комп'ютерно-математичне моделювання є основою для космічних досліджень та освоєння космосу.

1. Симуляція польотів – моделі забезпечують точне планування траєкторій польотів космічних апаратів, враховуючи вплив гравітації, магнітних полів та інших факторів. Це дає змогу уникнути помилок та знижує вартість місій.

2. Створення роботизованих систем – КММ застосовується для розробки роботів-дослідників, які можуть працювати в умовах екстремального холоду, радіації або низької гравітації, наприклад, на Марсі.

3. Підготовка астронавтів – моделі використовуються для симуляції умов перебування в космосі, зокрема впливу мікрогравітації на організм людини, що допомагає краще підготувати астронавтів до тривалих місій [4].

Висновки. Перспективи розвитку комп'ютерно-математичного моделювання є надзвичайно широкими. Інтеграція штучного інтелекту, використання квантових обчислень і впровадження хмарних технологій відкривають нові горизонти для науки та інженерії. Ці напрями дають змогу зробити КММ більш доступним, точним та ефективним, сприяючи прогресу в багатьох галузях людської діяльності.

Список використаних джерел

1. Berman F., Fox G., Hey A. J. G. Grid Computing: Making the Global Infrastructure a Reality. Wiley-Interscience, 2003. 1060 p.

2. Nielsen M. A., Chuang I. L. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge: Cambridge University Press, 2010. 702 p.

3. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 3rd edition, Pearson Education, 2010. 1152 p.

4. Peng L., Bao L., Huang M. Application of Matlab / Simulink Software in Physics. *High Performance Networking, Computing, and Communication Systems.* Springer Berlin Heidelberg, 2011. P. 140–146.

УДК 338.27:004.89

Іваненко А. В., здобувач вищої освіти,
Січко Т. В., канд. техн. наук, доцент
кафедри інформаційних технологій

РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРОГНОЗУВАННЯ СПОЖИВЧОГО ПОПИТУ МЕТОДАМИ ШТУЧНОГО ІНТЕЛЕКТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Прогнозування попиту – це процес оцінки майбутнього попиту на товар або послугу, який дає підприємствам змогу передбачити кількість продукції чи послуг, які споживачі можуть придбати у певний період часу [1]. Точність прогнозування є критично важливою для ефективного управління бізнесом, оскільки вона дає змогу правильно планувати виробничі процеси, розподіляти ресурси, оптимізувати логістику та здійснювати стратегічні інвестиції. Неправильна оцінка попиту може призвести до надлишку запасів, що тягне за собою зайві витрати на зберігання, або до дефіциту продукції, що знижує задоволення споживачів і втрачає потенційний дохід. Тому точне прогнозування попиту є

основою для прийняття обґрунтованих рішень у плануванні виробництва та управлінні запасами.

Метою дослідження є застосування методів машинного навчання для прогнозування попиту, а саме прогнозування часових рядів.

Машинне навчання – це сфера штучного інтелекту, що фокусується на розробці алгоритмів і моделей, здатних навчатися на основі даних і покращувати свою роботу без явного програмування [2]. В основі машинного навчання лежить ідея, що системи можуть самостійно виявляти патерни та структури в даних, адаптуватися до нової інформації та приймати рішення на основі набутих знань. Передовими розробками у сфері машинного навчання є нейронні мережі, які поступово витісняють більш традиційні рішення.

Нейромережі – це моделі, натхненні біологічними нейронними мережами, що складаються зі штучних нейронів – структурних одиниць, які приймають дані, виконують над ними прості обчислення та передають цю інформацію далі [2, 3]. Нейронні мережі складаються з шарів нейронів: вхідного для прийому даних, прихованих для їх обробки та вихідного для видачі результатів. Для прогнозування часових рядів застосовують такі архітектури нейромереж:

- RNN (Recurrent Neural Network) – є архітектурою нейронних мереж, яка ефективно працює з послідовними даними, як-от часові ряди. Особовості RNN – це наявність зворотних зв’язків між прихованими шарами мережі, що дає змогу їм використовувати інформацію з попередніх кроків. Рекурентні зв’язки допомагають зберігати інформацію з попередніх кроків обробки даних, що робить цю архітектуру здатною до аналізу залежностей між елементами послідовності;
- GRU (Gated Recurrent Unit) – тип рекурентної нейронної мережі, що вирішує проблеми довготривалої залежності у послідовностях. Окрім стандартних вхідних, прихованих та вихідних шарів, мережа GRU також має дві затворні структури: затвор оновлення (update gate) та затвор скидання (reset gate), які визначають, яку інформацію зберігати або ігнорувати. Затвор оновлення контролює, скільки нової та старої інформації зберігати, а затвор скидання – скільки даних з минулого забути. Це забезпечує ефективну роботу з послідовними даними, наприклад, у прогнозуванні часових рядів;
- LSTM (Long Short-Term Memory) – архітектура нейронної мережі, що також була створена для вирішення проблеми довготривалої залежності у послідовних даних, з якою стикалися звичайні рекурентні нейронні мережі (RNN). LSTM використовують спеціальні механізми для збереження важливої інформації на тривалий період часу, що робить їх ефективними у задачах обробки послідовностей, як-от прогнозування часових рядів;
- CNN (Convolutional Neural Network) – архітектура, спочатку розроблена для обробки зображень, яка також використовується для аналізу сигналів і часових рядів. Вона складається зі вхідного шару, згорткових шарів (convolution layers), шару активації, агрегувальних шарів (pooling layers), повнозв’язних (fully connected layers) і вихідного шару. Шари згортки є основними шарами цієї мережі, у яких до вхідних даних застосовуються фільтри, що дає змогу виділяти локальні ознаки. Кожен фільтр зчитує невеликий набір даних і застосовує операцію згортки для виявлення патернів у цьому локальному наборі.

У випадку часових рядів використовуються одновимірні згортки (1D Convolution), де фільтри переміщаються уздовж часової осі, що допомагає моделі вивчати послідовні залежності. Агрегувальні шари зменшують розмірність даних, а повнозв'язні шари узагальнюють ознаки та приймають рішення, наприклад, прогнозують наступне значення.

До того ж новітніми розробками в цій сфері є гібридні нейронні мережі, що поєднують у собі ознаки декількох типів нейромереж або нейронної мережі з іншими, більш традиційними методами машинного навчання. Така інтеграція різних моделей дає змогу використовувати їх сильні сторони для обробки різноманітних типів даних і врахування складних взаємозв'язків, що робить прогнозування попиту більш ефективним і точним. Прикладами гібридних моделей, використовуваних під час прогнозування часових рядів, є:

- LSTM-CNN – гібридна мережа, що поєднує переваги CNN і LSTM. Ця архітектура спочатку використовує CNN для виявлення локальних, короткотермінових патернів у даних [4]. CNN працює шляхом застосування фільтрів до послідовностей даних, виявляючи важливі характеристики (наприклад, різкі зміни чи локальні піки). Після цього вихід CNN передається до LSTM, яка відома своєю здатністю запам'ятовувати довготривалі залежності у послідовностях. LSTM має механізми збереження та забування інформації, що дає їй змогу зберігати релевантні патерни у довгостроковій перспективі і одночасно позбавлятися зайвих. Така комбінація допомагає враховувати як короткострокові, так і довгострокові характеристики, що підвищує точність прогнозів, зокрема за високої мінливості або сезонності даних.
- ES-RNN – гібридна модель, що об'єднує експоненційне згладжування (ES) і рекурентну нейронну мережу (RNN) [5]. Експоненційне згладжування – це один із традиційних методів машинного навчання для прогнозування часових рядів, який базується на використанні зваженого середнього попередніх значень ряду, де ваги експоненційно зменшуються в часі. Ця модель розроблена для того, щоб обробляти як короткострокові, так і довгострокові залежності в часі, що є ключовим фактором для точного передбачення тенденцій у часових рядах. Експоненційне згладжування використовується для виділення основних трендів і сезонних компонентів у даних. Завдяки своїм властивостям експоненційне згладжування здатне адаптивно враховувати зміни в трендах і циклах, зменшуючи вплив короткострокових коливань. З іншого боку, рекурентна нейронна мережа добре підходить для обробки складних часових залежностей і взаємозв'язків між попередніми та поточними значеннями в ряді. RNN здатна запам'ятовувати інформацію про минулі стани завдяки своїй рекурентній структурі, що дає змогу їй краще моделювати послідовні процеси. Ця комбінація допомагає ефективно фільтрувати дані та точно прогнозувати складні часові ряди.

Отже, було розглянуто різноманітні методи для прогнозування попиту, а саме сучасні підходи на основі технологій нейронних мереж. Приділено увагу як поширеним архітектурам нейромереж, так і більш комплексним гібридним моделям, які є одним із новітніх шляхів розвитку сфери машинного навчання. Розробка нових більш ефективніших рішень дасть змогу значно підвищити точність прогнозування попиту, що є критичним для успішної стратегії управ-

ління підприємств. Це допоможе краще оптимізувати ресурси, знижувати витрати на зберігання запасів, покращувати логістичні процеси та забезпечити більшу гнучкість у відповідь на зміни ринку для підприємств.

Список використаних джерел

1. Jr C. C. W., Chase C. W. Demand-Driven Forecasting: A Structured Approach to Forecasting. Wiley Sons, Limited, John, 2015.

2. Huang C., Petukhina A. Applied time series analysis and forecasting with python. Springer International Publishing AG, 2022.

3. Method of neural network detection of anomalies in data of waste-free production audit / Т. Нескородєва, Є. Федоров, А. Нескородєва, Т. Січко, П. Римар. *Computer systems and information technologies.* 2021. № 2. С. 20–32.

4. Aguiar-Pérez J. M., Pérez-Juárez M. Á. An insight of deep learning based demand forecasting in smart grids. 2023. Т. 23. № 3. С. 1467.

5. Smyl S., Dudek G., Pełka P. ES-dRNN: a hybrid exponential smoothing and dilated recurrent neural network model for short-term load forecasting. *IEEE transactions on neural networks and learning systems.* 2023. P. 1–13. DOI: 10.1109/tnnls.2023.3259149 (date of access: 15.11.2024).

УДК 004.62:519.6;681.586

Канюка Р. Ю., здобувач вищої освіти,
Роштейн О. П., д-р техн. наук, професор,
професор кафедри інформаційних технологій

МАТЕМАТИЧНА ОБРОБКА ОТРИМАНИХ ДАНИХ УЛЬТРАЗВУКОВОГО ДАТЧИКА

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Звукові сигнали мають широке застосування в різних сферах. Одним з найбільш поширеного використання є звукова локалізація, яка дає змогу визначити місце та напрямок джерела звуку шляхом аналізу сигналу та часу його повернення. Ця методика активно використовується в наукових дослідженнях для вивчення атмосферних явищ, як-от блискавки та турбулентність, а також у геофізичних дослідженнях, зокрема для аналізу сейсмічної активності.

У промисловості звукова локалізація слугує для неінвазійного обстеження матеріалів та конструкцій, таких як-от труби, мости та будівлі. Завдяки простоті цієї технології можна легко створити прототип, використовуючи базові знання, оскільки для реалізації достатньо мікрофона або подібного пристрою. Використовуючи швидкість звуку в повітрі та виміряний час між моментом випромінювання та отриманням відбитого сигналу, можна обчислити відстань до об’єкта.

Це можна виразити через формулу (1.1), що базується на методі часу польоту:

$$d = \frac{vt}{2}. \qquad (1.1)$$

Для вирішення проблеми шумів застосовують методи усереднення (передбачення) – *метод Калмана.* Метод дає змогу зменшити похибку в точності вимірювань відстані до дистанції акустичного джерела.

Основний текст. У досліджені використовуватиметься готовий прототип (рис. 1) для отримання даних та їх подальшого покращення.

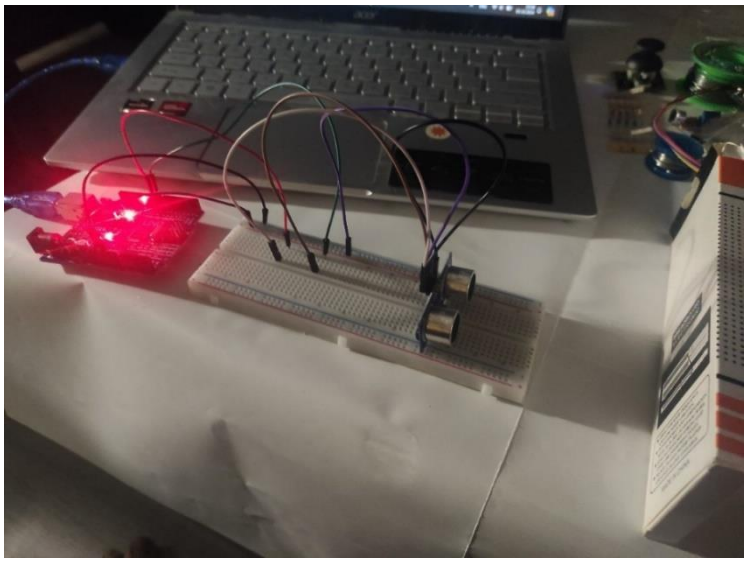


Рисунок 1 – Прототип ультразвукового датчика на основі Arduino HC-SR04

Ультразвуковий датчик працює за допомогою вимірювання часу, який необхідний звуковій хвилі, щоб пройти від датчика до об’єкта та повернутися назад. **Основною проблемою датчика є реакція на рух.** Компонентами ультразвукового датчика є випромінювач, який генерує ультразвуковий сигнал, і приймач, що фіксує відбитий сигнал. Ультразвукові датчики використовуються в різних галузях, як-от робототехніка, автомобільна промисловість (для пакувальних систем), системи безпеки та автоматизації, для вимірювання відстаней до об’єктів, виявлення перешкод та визначення положення об’єктів.

Він випромінює ультразвукові хвилі з частотою зазвичай приблизно 40 кГц, які відбиваються від об’єкта, а потім повертаються до приймача, що дає змогу виміряти відстань до об’єктів. Датчик має хороший діапазон вимірювання (від 2 см до 400 см) і прийнятну точність (зазвичай ±3 мм). HC-SR04 простий у підключенні до мікроконтролерів, як-от Arduino, і знайомий багатьом інженерам через свою доступність та низьку вартість.

Швидкість роботи датчика становить 20–30 мілісекунд на одне вимірювання, що дає змогу отримувати 35–50 вимірювань відстані за секунду.

Як було вже описано, **Метод Калмана** використовує математичну модель процесу для прогнозування наступного стану системи на основі попередніх даних. Це означає, що він може передбачити, які значення мають бути, і уникнути впливу шумів. Загальна формула представлення (1.2):

$$\hat{x}k, k = \frac{1}{k}\sum_{i=0}^{k-1} x_i \,. \qquad (1.2)$$

Висновки. За допомогою Arduino та фільтрації Калмана здійснимо рух на шляху датчика з певним коливанням на місці. Результат тестування дав змогу підвищити точність. Відображений результат на рис. 2 допомагає нехтувати постійними коливаннями тіла, що рухається, щоб знайти середню відстань.

Рисунок 2 – Графік відображення отриманих даних по відстані та обробленого результату

Подальше покращення продукту та додавання сервоприводу, другого датчика та розвинутого фільтру Калмана дасть змогу покращити точність приладу та реагувати на рух краще.

Список використаних джерел

1. Дніпровський державний технічний університет. URL: http://pzs.dstu.dp.ua/DataMining/kalman/index.html (дата звернення: 24.10.2024).

2. Дончак О. В. Спосіб визначення просторового положення об’єктів на основі алгоритму TDOA: магістерська дис.: 123 Комп’ютерна інженерія (Комп’ютерні системи та компоненти). Київ, 2018. 109 с.

3. Бондаренко А. Просторова локалізація акустичних подій в електронній музиці. Матеріали II Науково-практичної конференції: *Конструювання майбутнього* (м. Вінниця, 25 вересня 2019 р.) 2020. 26–30 с.

4. Raman Aj. Навчальна робота реалізації акустичної системи за допомогою Arduino Uno та HC-SR04. URL: https://www.instructables.com/Ultrasonic-Mono-Pulse-Tracker/ (дата звернення: 24.10.2024).

УДК 004.9

Кизь Н. А, здобувач вищої освіти,
Луценко А. В., доктор філософії з математики, старший викладач,
в. о. завідувача кафедри прикладної математики та кібербезпеки

МЕТОДИ ОБРОБКИ ТА АНАЛІЗУ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Анотація. У роботі досліджено методи обробки та аналізу даних, що є ключовими етапами в науковій, технічній та прикладній діяльності; основні завдання аналізу, зокрема

перевірку гіпотез, узагальнення результатів та формулювання висновків. Підкреслено важливість точності, достовірності та систематизації даних для забезпечення ефективності досліджень. Представлено приклади практичного застосування, які демонструють роль даних у прийнятті рішень та впровадженні інновацій.

Ключові слова: аналіз даних, інформація, статистичні методи, дослідження, технології.

Вступ. У сучасному світі зростає кількість ресурсів, від яких залежать стратегічні рішення в різних сферах діяльності, як-от наука, техніка та бізнес. Оскільки інформаційні потоки невпинно зростають, виникає потреба в якісному зборі, обробці та аналізі даних для визначення закономірностей та прийняття оптимальних рішень. Методи обробки та аналізу даних дають змогу покращити якість опрацювання інформації для підвищення ефективності та конкурентоспроможності організацій.

Метою роботи є дослідження методів обробки та аналізу даних.

Основний текст. Обробка даних є процесом, що спрямований на перетворення необробленої інформації в структуровану, яку можна використовувати для аналізу або проведення практичних досліджень. На початковому етапі отримані дані потребують упорядкування та організації, щоб зробити їх придатними для подальшої роботи [1, с. 93–94].

Одним із найпоширеніших способів обробки даних є створення електронних таблиць для впорядкування. Для цього використовуються програми Excel, Google Таблиці або інші спеціалізовані інструменти, наприклад, програмування через Python, що значно спрощують роботу з великими обсягами інформації.

Результатом обробки даних є створення основи для подальшого аналізу, що є ключовим видом дослідницької роботи, який спрямований на виявлення закономірностей, характеристик і тенденцій розвитку досліджуваного об’єкта. Його основна мета полягає в систематичному дослідженні даних у такий спосіб, щоб отримати ґрунтовні висновки. Тому процес аналізу даних включає відбір та розрахунок отриманих показників, перевірку гіпотез, демонстрацію результатів дослідження та формулювання висновків. Аналіз у цьому випадку також включає перевірку достовірності зібраної інформації та її актуальність, завдяки чому забезпечується послідовність, точність, обґрунтованість та логічна узгодженість усіх етапів досліджень.

Наступним етапом є визначення характеристик і показників, які допомагають оцінити об’єкт або процес з об’єктивного та суб’єктивного погляду.

Доведення та перевірка гіпотез є невід’ємним складником аналітичної роботи: гіпотези проходять через етап статистичного або теоретичного підтвердження, що дає змогу обґрунтувати їх.

Узагальнення результатів також є важливою частиною, адже воно дає можливість сформулювати висновки, які відображають закономірності або характеристики об’єкта дослідження [1; 2, с. 36].

Важливим аспектом є забезпечення точності та надійності отриманих даних, тому необхідно звернути увагу на способи їх інтерпретації, узагальнити результати й визначити шляхи вирішення проблем отриманих результатів.

Наприклад, аналіз даних є ключовим аспектом будь-якої програми соціологічного дослідження, де основні завдання включають визначення типу та обсягу необхідної інформації, а також вибір методів та інструментів для її збору, реєстрації, обробки та перетворення.

Висновки. Методи обробки та аналізу даних є потужними інструментами для отримання цінної інформації з різноманітних джерел. Вони сприяють прийняттю раціональних рішень, розвитку інновацій і дають змогу вирішувати складні завдання в різних галузях. У бізнесі вони допомагають оптимізувати маркетингові кампанії, прогнозувати продажі та сегментувати клієнтів; у медицині – використовуються для діагностики захворювань та визначення ефективності лікування. Однак робота з даними супроводжується викликами, до яких відносять недостовірність даних, великий обсяг інформації, що потребує значних ресурсів, і питання їх безпеки, зокрема захисту персональних даних. Подолання цих викликів вимагає впровадження нових технологій та удосконалення наявних методів.

Сучасний світ вимагає все більшого використання цих методів, що підкреслює їх важливість для науки, бізнесу та суспільства загалом.

Список використаних джерел

1. Методи обробки та інтелектуального аналізу даних з використанням штучних імунних систем / А. С. Бурда, М. А. Прудіус, Я. Г. Стефанюк, О. О. Фомічов. *Системи управління, навігації та зв'язку.* Полтава, 2024. № 3. С. 93–96.

2. Джога М. О., Ніколюк П. К. Методи обробки і аналізу даних. *Комп'ютерні технології обробки даних*. Секція 1. Методи обробки і аналізу даних, 2023. С. 35–37.

УДК 004.422.5

Курбангалієв Д. М., здобувач вищої освіти,
Поремський Ю. В., канд. техн. наук,
старший викладач кафедри інформаційних технологій

АЛГОРИТМИ МАШИННОГО НАВЧАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритми машинного навчання стали основою сучасного цифрового світу, даючи змогу вирішувати складні завдання, які були неможливі кілька десятиліть тому. Одним із найбільш яскравих прикладів їх впливу є нейронні мережі та їх застосування у комп'ютерному зорі. Нейронні мережі особливо глибоко демонструють неймовірну ефективність у розпізнаванні об'єктів на зображеннях, що допомагає створити системи для медичної діагностики, автономного водіння та навіть мистецької творчості. Оскільки машинне навчання є важливим складником багатьох сучасних систем, знання алгоритмів та методів його реалізації є необхідними не лише для фахівців, а й для широкої аудиторії, зокрема і школярів та студентів [1].

Машинне навчання (Machine Learning) – це підгалузь штучного інтелекту, що розробляє алгоритми, які дають змогу комп'ютерам вивчати закономірності в даних і використовувати ці знання для прийняття рішень. Процес машинного навчання включає кілька етапів:

1. Збір даних: дані збираються з різних джерел (системи, сенсори, інтернет).

2. Попередня обробка даних: очищення та підготовка даних для подальшого аналізу.

3. Навчання моделі: на основі даних модель починає формувати правила, які використовуються для прогнозування або прийняття рішень.

4. Оцінка моделі: перевірка моделі на тестових даних для оцінки її точності та ефективності.

5. Використання та розгортання: готову модель застосовують у реальних задачах [2].

Машинне навчання розподіляється на кілька типів залежно від алгоритмів машинного навчання:

1. Кероване навчання (Supervised Learning) – це підхід у машинному навчанні, коли модель тренується на основі мічених даних.

1.1 Лінійна регресія. Лінійна регресія є одним з простих алгоритмів для прогнозування числових значень. Вона будує лінію (або гіперплощину в багатовимірному просторі), яка мінімізує помилку між передбаченими та фактичними значеннями.

1.2 Логістична регресія. Логістична регресія – це статистичний метод, що використовується для бінарної класифікації. Замість лінії логістична регресія використовує функцію логістичної (сигмоїдної) для прогнозування ймовірностей того, до якого з двох класів належить спостереження.

1.3 Дерево рішень. Дерево рішень – це структура, що розбиває дані на різні класи за допомогою послідовних умов. Кожен вузол дерева – це тест або перевірка, що розділяє дані на підгрупи, а кожен листовий вузол – це передбачення або результат для певної групи.

1.4 Алгоритм K-ближчих сусідів (KNN). Алгоритм K-ближчих сусідів працює за принципом «схожих елементів». Коли потрібно класифікувати новий елемент, алгоритм визначає найближчих K сусідів у навчальних даних і призначає клас на основі більшості голосів.

2. *Некероване навчання (Unsupervised Learning).* У некерованому навчанні модель працює з даними без міток. Метою є виявлення структур чи патернів в даних, як-от групи або залежності між ознаками.

2.1 Метод головних компонент (PCA). Метод головних компонент (PCA) використовується для зменшення розмірності даних. Це дає змогу зберегти основну інформацію, зменшуючи кількість ознак, які можуть бути використані для подальшого аналізу або моделювання. PCA шукає нові осі (головні компоненти), які максимізують дисперсію в даних.

2.2 Кластеризація K-середніх. Кластеризація K-середніх – це популярний алгоритм для групування схожих елементів у кластери. Користувач визначає кількість кластерів (K), і алгоритм намагається знайти центри кластерів так, щоб елементи всередині кожного кластера були якомога більш схожими.

2.3 Алгоритм самоорганізуючих карт (SOM). Самоорганізуючі карти (SOM) – це тип нейронної мережі, що використовує метод кластеризації для перетворення багатовимірних даних у більш зрозумілу 2D- або 3D-форму.

3. *Напівкероване навчання (Semi-supervised Learning).* Напівкероване навчання є комбінованим підходом, коли навчання проводиться з використанням як мічених, так і немічених даних. Алгоритм намагається використовувати додаткову інформацію з немічених даних, щоб покращити точність моделі.

4. *Машинне навчання з підкріпленням (Reinforcement Learning).* Машинне навчання з підкріпленням відрізняється від інших типів навчання тим, що агент вчиться через взаємодію з середовищем. Він отримує винагороду або покарання залежно від того, як добре виконана дія, і з часом намагається максимізувати суму отриманих винагород. Класичний приклад – це Q-навчання, де агент поступово вивчає стратегію для досягнення максимальної винагороди [4].

Машинне навчання активно застосовується в багатьох сферах:

Соціальні мережі. Машинне навчання дає змогу платформам, як-от Facebook, Instagram і LinkedIn, аналізувати активність користувачів і на основі цієї інформації пропонувати відповідних друзів, контент або рекламу.

Охорона здоров'я. Алгоритми можуть аналізувати зображення медичних сканувань, як-от рентгенівські знімки або МРТ, для виявлення патологій, наприклад, ракових пухлин або захворювань ока.

Фінанси. У фінансовій сфері машинне навчання застосовується для оцінки кредитного ризику, де системи аналізують фінансову історію клієнта, його платіжну здатність і прогнозують ймовірність неплатежу.

Ритейл (торгівля). Машинне навчання в ритейлі, наприклад, у компаніях Amazon, eBay, використовується для персоналізації рекомендацій товарів.

Транспорт і транспортна логістика. Машинне навчання активно використовується у сфері транспорту для створення систем автоматизованого управління дорожнім рухом. Безпілотні автомобілі – це яскравий приклад використання технологій машинного навчання в транспорті.

Інтернет речей (IoT). Інтернет речей є ще однією сферою, де машинне навчання дає змогу зібрати дані з великої кількості сенсорів і пристроїв, як-от смарт-годинники, домашні термостати, розумні камери спостереження тощо.

Розваги та медіа. Машинне навчання також застосовується в індустрії розваг для персоналізації контенту. Алгоритми рекомендують фільми, музику, книги або відео на основі попередніх вподобань користувача [3].

Отже, машинне навчання є не тільки важливим компонентом сучасних технологій, але й потужним інструментом для трансформації численних індустрій. Його здатність автоматизувати складні процеси, обробляти великі обсяги даних та робити прогнози має величезний потенціал. Вивчення основ машинного навчання дає змогу студентам і школярам розвивати критичне мислення і готуватися до роботи в інноваційних сферах. Просте освоєння алгоритмів, як-от KNN або дерево рішень, створює хорошу основу для розуміння складніших концепцій і сприяє підготовці до кар'єри в технологічних компаніях [1].

Список використаних джерел

1. Alpaydin E. Introduction to Machine Learning. Book. The MIT Press. 2014. 600 p.
2. Bishop C. M. Pattern Recognition and Machine Learning. Book. Springer. 2006. 738 p.
3. Goodfellow I., Bengio Y., Courville A. Deep Learning. Book. MIT Press. 2016. 775 p.
4. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning.Book. Springer 2009. 745 p.
5. Mitchell T. M. Machine Learning. Book. McGraw-Hill. 1997. 414 p.
6. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach. Book. Prentice Hall. 2010. 1152 p.

УДК 004.9

Лавренюк Б. В., здобувач вищої освіти,
Потапова Н. А., канд. екон. наук, доцент,
доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ АНАЛІЗУ ДАНИХ ДЛЯ ПРОГНОЗУВАННЯ ТА ВИЯВЛЕННЯ ТРЕНДІВ У СВІТІ КІНЕМАТОГРАФІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасний кінематограф все більше інтегрується з технологіями, що дає змогу глибше розуміти вподобання аудиторії та передбачати майбутні тенденції. Одним із найпотужніших інструментів для цього є аналіз даних. Використання великих обсягів інформації дає змогу не лише прогнозувати успіх кінопроєктів, але й оперативно реагувати на зміни у запитах глядачів. У цьому контексті розкриття можливостей аналізу даних у кіноіндустрії набуває стратегічної значущості.

З огляду на те, що стримінгові сервіси, соціальні мережі та онлайн-рецензії формують глобальний кіноринок, кінокомпанії зіштовхуються з викликом не лише створення якісного продукту, а й конкурентної боротьби за увагу глядача. Аналіз даних допомагає не лише передбачити успішність майбутніх проєктів, але й знижує фінансові ризики за допомогою обґрунтованого прийняття рішень. У світі, де смаки аудиторії змінюються надзвичайно швидко, здатність виявляти тренди стає визначальним чинником успіху.

Аналіз даних став одним із ключових інструментів у світі кінематографії для виявлення трендів і прогнозування успіху проєктів. Провідні гравці індустрії активно використовують сучасні технології аналізу, як-от машинне навчання, обробка природної мови (NLP), кластеризація даних і алгоритми глибокого навчання, щоб зрозуміти вподобання аудиторії та адаптуватися до змін на ринку.

Netflix, один із лідерів стримінгового ринку, активно застосовує алгоритми машинного навчання для аналізу даних про глядачів. Система аналізує мільярди годин переглядів, включно з даними про час перегляду, пропуски, паузи та повторні перегляди. Наприклад, під час розробки серіалу «Дивні дива» (Stranger Things) компанія враховувала популярність жанру наукової фантастики серед молодої аудиторії, що була підтверджена аналізом історичних

переглядів та обговорень у соціальних мережах. Використовуються інструменти Apache Spark для обробки великих даних і AWS Redshift для зберігання даних.

Disney+ застосовує аналітичні інструменти Tableau та Snowflake для сегментації аудиторії та визначення потенційно успішних тематик. Наприклад, аналіз даних показав, що аудиторія в Південній Америці має особливий інтерес до локальних героїв та міфології. Це підштовхнуло Disney до створення проєктів, натхненних місцевою культурою, як-от серіал «Союз захисників Латинської Америки». Залучення даних про взаємодії глядачів із контентом також дає змогу адаптувати маркетингові кампанії до кожного регіону.

Amazon Prime Video інтегрує технології машинного навчання через свою платформу AWS SageMaker. Цей інструмент використовується для аналізу відгуків глядачів у реальному часі. Наприклад, перед випуском серіалу «The Boys» було проведено аналіз коментарів на Reddit і Twitter за допомогою обробки природної мови (NLP). Це допомогло визначити, які аспекти супергеройського жанру є найбільш цікавими для цільової аудиторії, як-от темні та іронічні сюжети. На основі цих даних маркетингові матеріали були скориговані для акцентування на брутальності та антигеройських характеристиках персонажів.

Warner Bros. активно використовує платформу IBM Watson для аналізу глядацьких уподобань та формування трейлерів. За допомогою технологій Watson Visual Recognition і Watson Natural Language Understanding компанія аналізує візуальні та текстові відгуки про попередні трейлери. Наприклад, у випадку з фільмом «Джокер» Watson виявив, що аудиторія віддає перевагу глибоким психологічним темам і менш звертає увагу на екшен. Це вплинуло на монтаж трейлерів, які робили акцент на особистій драмі головного героя.

Соціальні медіа, зокрема **Twitter** та **Instagram,** стали важливими джерелами для аналізу трендів. За допомогою інструментів Google Cloud Natural Language API компанії аналізують тональність і зміст постів, щоб виявляти нові теми, які цікавлять глядачів. Наприклад, у випадку фільму «Барбі» студія Warner Bros. використовувала дані з соціальних мереж для розуміння реакції на перші тизери. Аналіз постів, проведений за допомогою NLP, показав, що глядачі з інтересом обговорюють поєднання ностальгії та сучасного гумору. Це дало змогу студії зосередитися на цих аспектах у подальших промоційних кампаніях.

YouTube використовує алгоритми кластеризації даних, щоб аналізувати вподобання користувачів і виявляти нові тренди. За допомогою Google BigQuery компанія відслідковує, які трейлери викликають найбільшу кількість переглядів, лайків і коментарів, щоб спрямувати рекламні бюджети на просування таких фільмів. Успіх трейлера до фільму «Аватар: Шлях води» став можливим завдяки даним, які свідчили про сильний інтерес до візуальних ефектів та екологічної тематики.

Загалом аналіз даних перетворюється на стратегічний інструмент у кіноіндустрії. Популярні сервіси використовують як класичні методи статистики, так і сучасні технології машинного навчання, штучного інтелекту та обробки великих даних. Це дає змогу не лише прогнозувати касові збори, але й формувати контент, який відповідає реальним очікуванням глядачів, виявляючи тренди ще на ранніх етапах їх формування. У майбутньому саме здатність ана-

лізувати та використовувати дані все більше визначатиме успіх кінокомпаній у глобальному масштабі.

Список використаних джерел

1. Kulkarni S. S., Vaishnavi M., Kusuma H. Predicting the Conceptual Appeal of Movies using Data Analytics. *International journal of engineering research & technology (IJERT).* Vol. 09, iss. 02 (February 2020).
2. Movie Industry Economics: How Data Analytics Can Help Predict Movies' Financial Success / P. C. Murschetz, C. Bruneel, J.-L. Guy, D. Haughton, N. Lemercier, M.-D. McLaughlin, K. Mentzer, et al. Nordic journal of media management. 2020. Vol. 1, № 3. P. 339–359.
3. The Role of Data Analytics in Cinema. URL: https://www.cloudthat.com/resources/blog/the-role-of-data-analytics-in-cinema

УДК 004.032.26:621.397.331

Лаптєва М. А., здобувачка вищої освіти,
Хмелівський Ю. С., асистент кафедри
інформаційних технологій

ВИЯВЛЕННЯ УПЕРЕДЖЕНЬ У ШІ ЗА ДОМОПОГОЮ СТАТИСТИЧНОГО НАВЧАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Штучний інтелект (ШІ) дедалі більше стає невід'ємною частиною процесів прийняття рішень у різних галузях – від повсякденних потреб, робочих питань до надважливих для держави сфер. Однак зі зростанням популярності систем штучного інтелекту пропорційно зростає занепокоєння щодо помилок і упереджень, які можуть виникнути під час роботи. Вони можуть виникати на різних етапах конвеєра машинного навчання, включно зі збором даних, попередньою обробкою, вибором функцій, навчанням моделі та розгортанням. Виявлення та пом'якшення цих упереджень має вирішальне значення для забезпечення справедливості, підзвітності та прозорості.

Знаходження упереджень передбачає поєднання статистичного аналізу, знань предметної області та критичного мислення. Використовуючи методи перевірки гіпотез, регресійного аналізу і метрики справедливості, статистичне навчання може допомогти оцінити, чи непропорційно впливають прогнози або класифікації ШІ-моделі на певні групи. У цій статті досліджується роль статистичного навчання у виявленні упередженості в ШІ, його методології та застосування.

Зміщення в моделях ШІ часто перетинаються з різними джерелами, що робить складним завдання їх детекції. Щоб розв'язати проблему, спочатку потрібно виявити її корінь.

По-перше, часто проблеми виникають з навчальними даними. Важливо ретельно переглянути їх на наявність будь-яких властивих упереджень, тому що якщо ці дані відображають наявні суспільні упередження, модель збереже їх і сприйме як щось регулярне. Наприклад, набори даних, що використовуються

в системах розпізнавання облич, можуть недопредставляти групи меншин, що призводить до зниження точності для цих груп населення. Тому вибір даних для навчання ШІ має бути зроблений ретельно.

По-друге, упередженість може бути вкорінена в процесі навчання моделі через алгоритмічний вибір, гіперпараметри або цілі оптимізації. Алгоритми можуть посилювати упередження під час оптимізації точності або інших показників продуктивності, нехтуючи міркуваннями справедливості. Це явище, відоме як «алгоритмічне упередження», може призвести до ненавмисної дискримінації.

І, по-третє, творці механізмів штучного інтелекту можуть вносити власні упередження у вибір дизайну та процеси оцінки [1].

Але у машинному навчанні існує деяка «Fairness Metrics», що в перекладі означає «Метрика неупередженості або чесності». Ось декілька пунктів звідти, які допоможуть уникнути упередженості:

1. Статистичний або демографічний паритет. Вимірює, чи рівномірно розподілені прогнози між групами.

How to Calculate Statistical Parity:

$$P(Outcome=1|Group=A)=P(Outcome=1|Group=B)$$

Рисунок 1 – Формула обрахунку статистичного паритету

2. Рівність шансів. Гарантує, що рівень помилок моделі є однаковим для різних груп.

How to Calculate Equality of Odds:

$$P(Outcome=1|Actual=0,Group=A)=P(Outcome=1|Actual=0,Group=B)$$
$$P(Outcome=1|Actual=0,Group=A)=P(Outcome=1|Actual=0,Group=B)$$

Рисунок 2 – Формула обрахунку рівності шансів

3. Калібрування. Перевіряє, чи збігаються прогнозовані ймовірності з фактичними результатами для всіх підгруп [2].

Після оцінки справедливості моделі наступним кроком є аналіз даних, щоб зрозуміти джерела упереджень. **Дослідницький аналіз даних (EDA)** є ключовим етапом, який допомагає виявити проблеми ще до навчання моделі. За допомогою візуальних інструментів, як-от гістограми або діаграми розсіювання, можна побачити, чи є в наборі даних недостатньо представлені групи. Описова статистика, як-от середнє значення або розподіл даних, дає уявлення про дисбаланс, а статистичні тести, наприклад, хі-квадрат або t-тести, допомагають визначити, чи є ці відмінності статистично значущими.

Коли упередження виявлені, потрібно збалансувати дані, щоб усі групи були належно представлені. Для цього використовують методи **надмірної ви-**

бірки або **недостатньої вибірки.** Надмірна вибірка, наприклад, за допомогою техніки SMOTE створює синтетичні дані для недостатньо представлених груп, збільшуючи їх кількість. Недостатня вибірка, навпаки, зменшує кількість точок у перепредставлених групах. Обидва підходи допомагають моделі навчатись на більш збалансованому наборі даних, зменшуючи ризик упередженості в результатах [3].

Ще однією важливою частиною є робота з ознаками, які можуть викликати упередженість у моделі. Наприклад, деякі ознаки можуть бути проксі для чутливих атрибутів, як-от раса чи стать. Поштовий індекс, наприклад, може приховувати соціально-економічний статус і впливати на прогнози моделі. Аналіз кореляції дає змогу виявити такі небажані зв'язки. Якщо виявлено, що певна ознака несе упередженість або мало впливає на якість прогнозу, її можна видалити чи трансформувати. Методи зменшення розмірності, як-от PCA (аналіз головних компонент), допомагають залишити лише ті ознаки, які мають значення для моделі.

Для зменшення упередженості під час навчання моделі використовуються статистичні методи **регуляризації.** Вони додають обмеження, які карають модель за надмірну залежність від чутливих ознак. Наприклад, якщо модель сильно спирається на проксі-ознаки, регуляризація зменшує їх вплив, допомагаючи зробити прогнози більш справедливими. Інтеграція обмежень справедливості в процес навчання гарантує, що боротьба з упередженнями є частиною самої розробки моделі, а не відбувається після її створення.

Після завершення навчання моделі статистичні інструменти дають змогу оцінити результати та перевірити, чи залишились у прогнозах залишкові упередження. Наприклад, **контрфактичний аналіз** дає змогу перевірити, як модель змінює свої прогнози для двох схожих випадків, які відрізняються лише в одному чутливому атрибуті (наприклад, стать чи раса). Якщо модель дає різні результати, це вказує на дискримінацію. До того ж перевірка калібрування допомагає впевнитися, що передбачені ймовірності відповідають реальним результатам у всіх групах [4].

У підсумку, статистичне навчання дає змогу не лише виявляти упередження, а й активно боротися з ними на кожному етапі – від аналізу даних і підготовки до навчання моделі та її оцінки. Це забезпечує створення більш справедливих і прозорих систем.

Список використаних джерел

1. Responsible AI Development: Bias Detection and Mitigation Strategies. Agiliway. URL: http://surl.li/lsvpjm (дата звернення 29.11.2024).

2. Fairness Metrics In Machine Learning. Aporia. URL: http://surl.li/ygmtib (дата звернення 30.11.2024).

3. Machine Learning Yearning. Andrew Ng. 2019. 118 p. (дата звернення 30.11.2024).

4. Interpretable Machine Learning: A Guide For Making Black Box Models Explainable. Christoph Molnar. 2022. 328 p. (дата звернення 30.11.2024).

УДК 004.4:519.22]-057.68 (043.2)

Левченко М. Р., здобувачка вищої освіти,
Хмелівський Ю. С., асистент кафедри
інформаційних технологій

АНАЛІЗ ВИЖИВАННЯ ПАСАЖИРІВ НА БОРТУ TITANIC ЗА ДОПОМОГОЮ DATASET У МОВІ R

Донецький національний університет імені Василя Стуса, м. Вінниця

Аналіз даних – ключовий інструмент у сучасному світі для розуміння складних систем і прийняття рішень. Titanic Dataset є класичним прикладом реального набору даних, що дає змогу вивчати різноманітні аспекти виживання пасажирів під час катастрофи відомого лайнера. Цей датасет містить інформацію про демографічні, соціальні та економічні характеристики пасажирів, що допомагає досліджувати вплив різних факторів, як-от стать, вік чи соціальний статус, на шанси вижити.

Мова R, як одна з провідних мов для аналізу даних і статистичного моделювання, ідеально підходить для такого дослідження. Завдяки потужному інструментарію для візуалізації, обробки та аналізу даних R дає змогу не лише отримати інсайти, але й побудувати ефективні моделі для прогнозування.

Titanic Dataset часто використовується як навчальний інструмент для опанування методів статистичного аналізу, візуалізації даних та алгоритмів машинного навчання. Він є зручним для початківців, але також пропонує багато можливостей для глибшого аналізу [1].

1. Завантаження датасету та огляд за допомогою матриці діаграми розсіювання.

```
read.csv("/Users/marina/Desktop/stat_navych/titanic.csv")
titanic <- read.csv("/Users/marina/Desktop/stat_navych/titanic.csv")

summary(titanic)
pairs(titanic[, c("Age", "Fare", "Pclass")])
```

Рисунок 1 – Завантаження датасету

За допомогою цієї діаграми можна побачити зв’язок між змінними Age, Fare та Pclass.

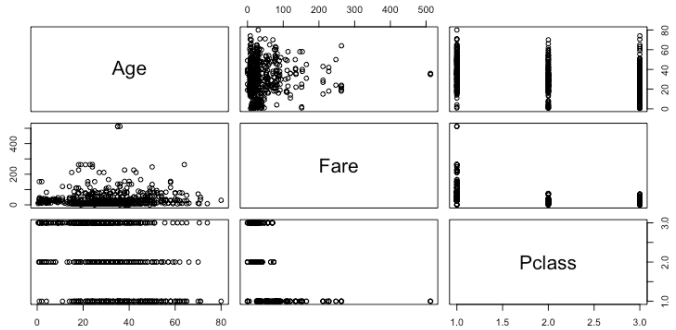


Рисунок 2 – Зв’язок між змінними

```
> summary(Titanic)
  PassengerId       Survived          Pclass          Name
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
 Median :446.0   Median :0.0000   Median :3.000   Mode  :character
 Mean   :446.0   Mean   :0.3838   Mean   :2.309
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
 Max.   :891.0   Max.   :1.0000   Max.   :3.000

     Sex                 Age            SibSp           Parch
 Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
 Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
 Mode  :character   Median :28.00   Median :0.000   Median :0.0000
                    Mean   :29.70   Mean   :0.523   Mean   :0.3816
                    3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
                    Max.   :80.00   Max.   :8.000   Max.   :6.0000
                    NA's   :177
    Ticket              Fare           Cabin             Embarked
 Length:891         Min.   :  0.00   Length:891         Length:891
 Class :character   1st Qu.:  7.91   Class :character   Class :character
 Mode  :character   Median : 14.45   Mode  :character   Mode  :character
                    Mean   : 32.20
                    3rd Qu.: 31.00
                    Max.   :512.33
```

Рисунок 3 – Загальний огляд датасету, його даних

2. Введемо додаткові зміни та візуалізуємо їх [2].

```
Expensive <- rep("Ні", nrow(titanic))
Expensive[titanic$Fare > median(titanic$Fare)] <- "Так"
Expensive <- as.factor(Expensive)
titanic <- data.frame(titanic, Expensive)

plot(titanic$Expensive, titanic$Survived, main = "Виживання залежно від вартості

par(mfrow = c(2, 2))
hist(titanic$Age, breaks = 10, main = "Розподіл віку")
hist(titanic$Fare, breaks = 15, main = "Розподіл вартості квитка")
hist(titanic$Parch, breaks = 6, main = "Розподіл кількості родичів")
hist(titanic$SibSp, breaks = 9, main = "Розподіл кількості братів/сестер")
```

Рисунок 4 – Візуалізація

За допомогою цього коду будуємо графік, який показує, як залежить виживання пасажирів від категорії вартості квитка («Дорогий» чи «Недорогий») [3].

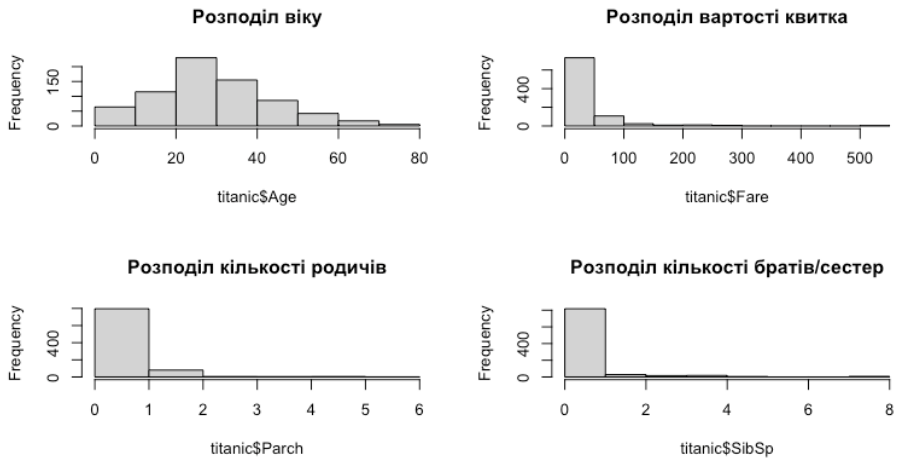


Рисунок 5 – Графіки залежностей

На основі графів можна зробити висновок, що середній вік пасажирів – 20–40 років, більшість квитків були недорогими, пасажири подорожували майже без родичів, вагома частина подорожувала подружжями [4].

3. Дослідження ключових факторів [3].

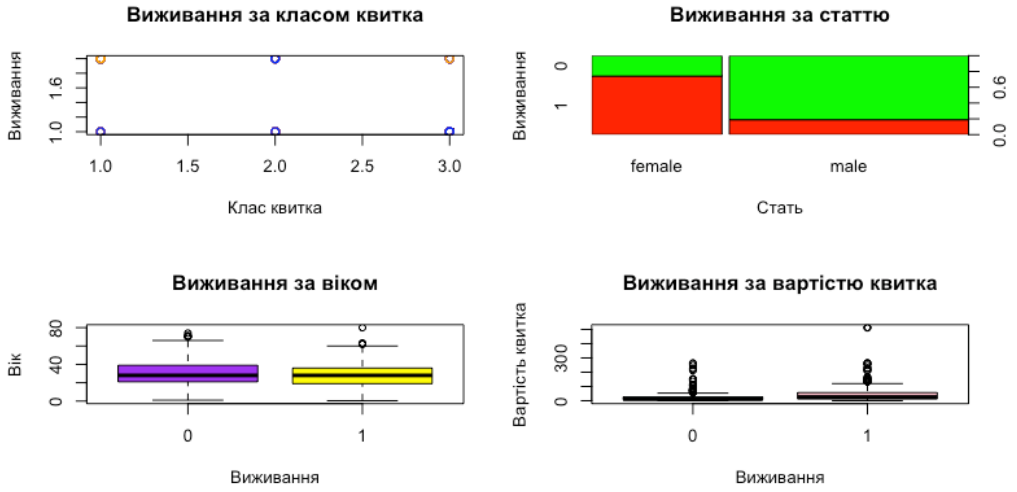


Рисунок 6 – Графік ключових факторів, які впливали на виживання

Уявімо, що ми опинилися на борту Титаніка, і ось наші шанси вижити за цією діаграмою:

Якщо у нас квитки першого класу, ми маємо більше шансів вижити. Також якщо наша стать жіноча, то шанси на виживання виростали ще більше, ніж чоловічої статі. Найбільшим везінням було б, якби ми були дітьми. І також найкраще, коли б квиток мав більшу вартість. За таких умов наші шанси на виживання були б дуже позитивними.

Результати цього аналізу показали, що мова R є потужним інструментом для проведення обробки даних, їх візуалізації та побудови статистичних висновків. Вивчення Titanic Dataset дало змогу не лише виявити закономірності в даних, але й продемонструвати важливість соціальних, економічних та демографічних факторів у виживанні людей під час надзвичайних ситуацій.

Отримані результати підтверджують ефективність використання статистичних методів для аналізу реальних подій, а також підкреслюють значення вивчення історичних даних для розуміння людської поведінки у кризових ситуаціях.

Список використаних джерел

1. DataSet. URL: https://www.kaggle.com/datasets
2. Освоюємо мову R. URL: https://uk.wikibooks.org/wiki/%D0%9E%D1%81%D0%B2%D0%BE%D1%8E%D1%94%D0%BC%D0%BE_R
3. Посібник КПІ з використанням мови R. URL: https://ela.kpi.ua/server/api/core/bitstreams/f78aa74c-7d8d-4c84-87eb-18a4aec57476/content
4. Data Science, лекція. URL: https://www.youtube.com/watch?v=DiKwo_hgFfM

УДК 004.422.5.

Ліхоткіна В. І., здобувачка вищої освіти,
Потапова Н. А., канд. екон. наук, доцент,
доцент кафедри інформаційних технологій

СУТНІСТЬ ТА СКЛАДНИКИ РЕКУРСИВНИХ АЛГОРИТМІВ У ПРОГРАМУВАННІ

Донецький національний університет імені Василя Стуса, м. Вінниця

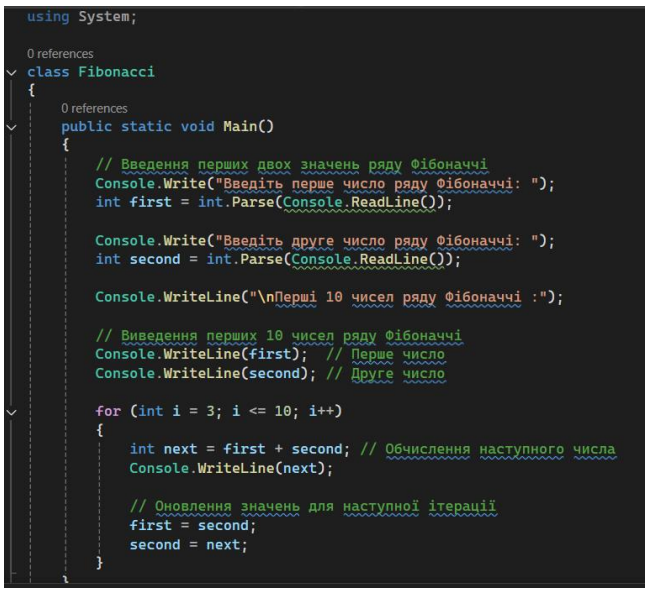
Термін «рекурсія» зустрічається у багатьох галузях науки і в більшості випадків має значення способу визначення множини об'єктів або функції через себе, з використанням раніше заданих значень. Рекурсія є потужним інструментом для розв'язку численних задач, опису природних явищ та станів. У галузі комп'ютерних наук рекурсія є важливою частиною реалізації програмного продукту. У програмуванні під рекурсією розуміють таку реалізацію, в якій метод використовує у своєму тілі виклик самого себе. Такі виклики називають рекурсивними. Рекурсію іноді важче реалізувати у вигляді програмного коду, але це додає більшої гнучкості для розв'язання задач [1].

За допомогою рекурсії реалізованого багато алгоритмів, які базуються на принципі повторення однієї і тієї самої дії певну кількість разів, поки завдання не буде виконане. За допомогою рекурсії можна реалізувати метод «розділяй та володарюй». *Складну* проблему ми розбиваємо на підзадачі, які є меншими копіями самої себе. Кожну підзадачу вирішуємо таким самим способом, поки не дійдемо до найпростішого випадку. Потім об'єднуємо результати всіх підзадач і отримуємо розв'язання початкової задачі. Є певний тип, задач які краще розв'язувати рекурсивним методом через те, що це значно спрощує завдання. До такого типу належать такі задачі:

1. Використання рекурсивних методів у розрахунку чисел Фібоначчі. Числа Фібоначчі – це послідовність чисел, у якій кожне наступне число є сумою двох попередніх. Починається вона з чисел 0 і 1. Для швидкого знаходження наступних чисел цієї послідовності зручно використовувати рекурсивний метод. Рекурсія в цьому випадку полягає в тому, що повторюється дія додавання двох попередніх членів цієї послідовності для знаходження наступного. Рекурсія в програмуванні реалізовується за допомогою циклів, на рисунку 1 це показано на прикладі задачі про числа Фібоначчі. За допомогою циклу задача виконується, доки не виконується умова.

Програма, що показана на рис. 1, за допомогою рекурсивного методу мовою програмування C# розв'язує задачу, і результат показаний на рис. 1.

Суть роботи коду в тому, що під час кожного повтору використовуються два числа – (first) та (second). Після проходження циклу і виконання дій у тілі циклу ці числа змінюють свої значення для проходження наступного значення – число (first) стає (second), число (second) стає (next).



```
using System;

0 references
class Fibonacci
{
    0 references
    public static void Main()
    {
        // Введення перших двох значень ряду Фібоначчі
        Console.Write("Введіть перше число ряду Фібоначчі: ");
        int first = int.Parse(Console.ReadLine());

        Console.Write("Введіть друге число ряду Фібоначчі: ");
        int second = int.Parse(Console.ReadLine());

        Console.WriteLine("\nПерші 10 чисел ряду Фібоначчі :");

        // Виведення перших 10 чисел ряду Фібоначчі
        Console.WriteLine(first);  // Перше число
        Console.WriteLine(second); // Друге число

        for (int i = 3; i <= 10; i++)
        {
            int next = first + second; // Обчислення наступного числа
            Console.WriteLine(next);

            // Оновлення значень для наступної ітерації
            first = second;
            second = next;
        }
    }
```

Рисунок 1 – Програмний код для реалізації задачі з числами Фібоначчі

2. Використання рекурсивних методів у задачах на графах. Рекурсивний метод для обходу бінарного дерева в прямому порядку (NLR – вузол, лівий піддерево, правий піддерево) працює на основі принципу «розділяй і володарюй», обробляючи кожну вершину дерева та її дочірні елементи по черзі [2].

3. Використання рекурсивних методів в алгоритмах сортування. Рекурсивні методи є основою роботи швидкого сортування та сортування злиттям, забезпечуючи ефективний розподіл задачі на підзадачі. У швидкому сортуванні рекурсія використовується для поділу масиву на дві частини відносно опорного елементу: всі менші елементи розташовуються зліва, а більші – справа. У сортуванні злиттям рекурсія використовується для поділу масиву на дві половини, які сортуються окремо. Після цього дві відсортовані половини зливаються у єдиний відсортований масив шляхом попарного порівняння елементів. Базовий випадок у сортуванні злиттям – це також масиви розміром один або порожні. Рекурсивна структура обох алгоритмів дає змогу ефективно розділяти та впорядковувати масиви, забезпечуючи логарифмічну глибину викликів. Рекурсивні методи часто використовують для реалізації завдань, розуміння рекурсії допомагає писати більш гнучкий та візуально кращий код, а також сприяє глибшому розумінню принципів програмування.

Список використаних джерел

1. Кублій Л. І. Алгоритми та структури даних. Основи алгоритмізації. Київ: Київський політехнічний інститут ім. Ігоря Сікорського, 2022. 528 с.

2. Грудзинський Ю. Є. Алгоритми та структури даних: навч. посіб. Київ: Київський політехнічний інститут ім. Ігоря Сікорського, 2022. 215 с.

УДК 004.422.63

Петров Д. Ю., здобувач вищої освіти,
Поремський Ю. В., канд. техн. наук,
старший викладач кафедри інформаційних технологій

ХЕШ-ТАБЛИЦІ: КОНЦЕПЦІЇ, РЕАЛІЗАЦІЯ ТА ПРИКЛАДИ ВИКОРИСТАННЯ У ПРОГРАМУВАННІ

Донецький національний університет імені Василя Стуса, м. Вінниця

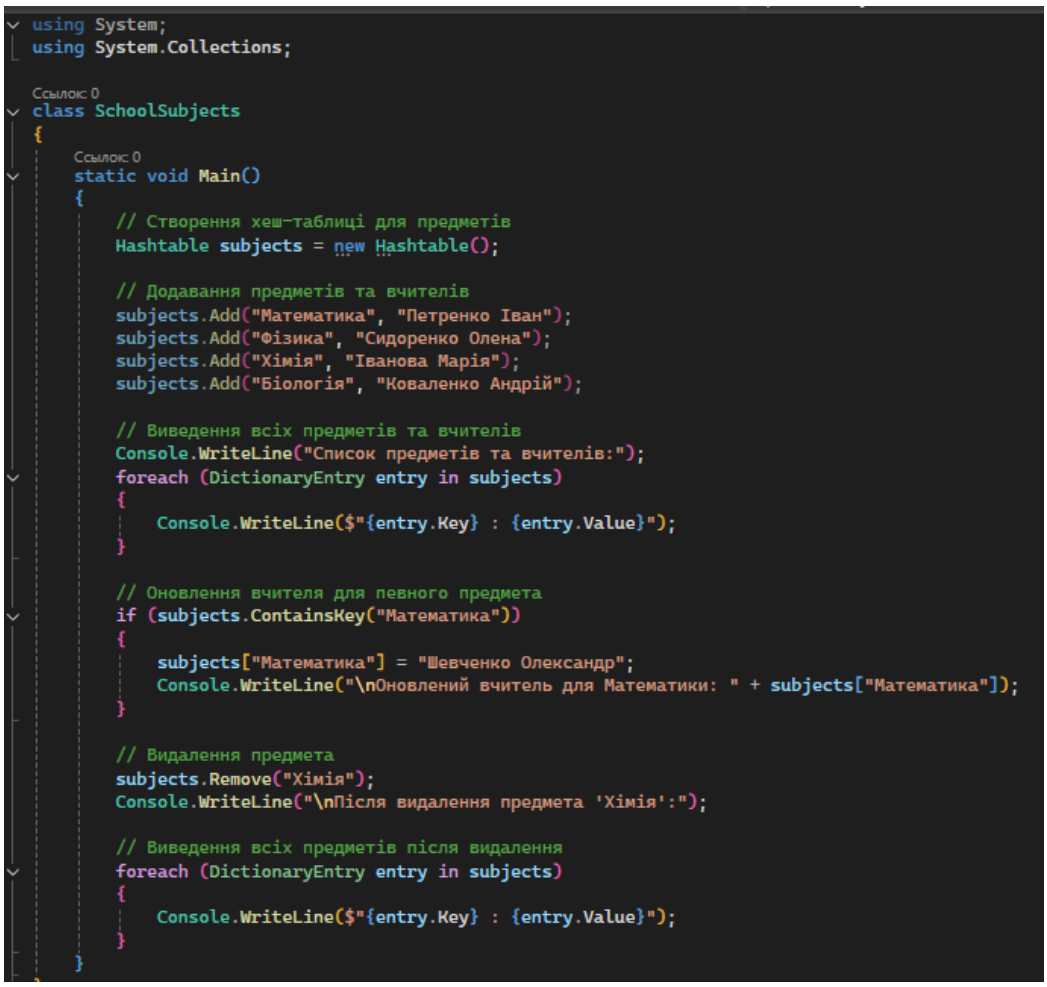
Хеш-таблиця – це структура даних, яка дає змогу швидко знаходити, вставляти і видаляти елементи з використанням хеш-функції для обчислення індексів у масиві або списку. Хеш-таблиці зазвичай використовуються для реалізації асоціативних масивів, також відомих як словники. Хеш-таблиці використовуються в багатьох сферах, включно з базами даних, кешуванням, пошуком дублікатів та обробкою великих обсягів інформації [1].

Основою роботи хеш-таблиць є хеш-функція – спеціальний алгоритм, який перетворює ключ у числовий індекс. Цей індекс використовується для швидкого доступу до елементу, що зберігається у відповідній позиції в масиві. Хеш-функція повинна бути ефективною, рівномірно розподіляти значення ключів по масиву та мінімізувати кількість конфліктів, що виникають у процесі обчислення. Завдяки використанню хеш-функції у багатьох випадках доступ до даних у хеш-таблиці здійснюється за постійний час, незалежно від кількості елементів [2].

Завдяки своїй універсальності хеш-таблиці широко застосовуються у різних областях інформатики та програмування. Вони є основою для реалізації словників, де кожен ключ відповідає певному значенню. До того ж хеш-таблиці використовуються для кешування даних, коли потрібно швидко зберігати і отримувати часто використовувану інформацію, як, наприклад, у вебкешах. Їх також активно застосовують у задачах аналізу текстів, зокрема для підрахунку частоти слів у тексті або для визначення унікальних слів у документі [3].

Аналогом хеш-таблиць у реальному житті можна вважати телефонні книги, де імена людей виступають у ролі ключів, а номери телефонів – значень. Є багато практичних прикладів хеш-таблиць, які використовуються в повсякденному житті. Популярним прикладом є бази даних імен користувачів і паролів. Щоразу, коли хтось реєструється на вебсайті за допомогою імені користувача та пароля, цю інформацію потрібно десь зберігати для подальшого пошуку. Хеш-функції широко використовуються в інформатиці та криптографії для різних цілей, зокрема і перевірки цілісності даних, зберігання паролів, цифрових підписів та індексації даних [4].

Приклад програмної реалізації простої хеш-таблиці для реалізації довідника предметів, які викладаються в школі, і відповідних вчителів наведено на рис. 1. Операції з додавання, оновлення, видалення предметів роблять код більш функціональним та реалістичним у контексті школи.



```
using System;
using System.Collections;

Ссылок: 0
class SchoolSubjects
{
    Ссылок: 0
    static void Main()
    {
        // Створення хеш-таблиці для предметів
        Hashtable subjects = new Hashtable();

        // Додавання предметів та вчителів
        subjects.Add("Математика", "Петренко Іван");
        subjects.Add("Фізика", "Сидоренко Олена");
        subjects.Add("Хімія", "Іванова Марія");
        subjects.Add("Біологія", "Коваленко Андрій");

        // Виведення всіх предметів та вчителів
        Console.WriteLine("Список предметів та вчителів:");
        foreach (DictionaryEntry entry in subjects)
        {
            Console.WriteLine($"{entry.Key} : {entry.Value}");
        }

        // Оновлення вчителя для певного предмета
        if (subjects.ContainsKey("Математика"))
        {
            subjects["Математика"] = "Шевченко Олександр";
            Console.WriteLine("\nОновлений вчитель для Математики: " + subjects["Математика"]);
        }

        // Видалення предмета
        subjects.Remove("Хімія");
        Console.WriteLine("\nПісля видалення предмета 'Хімія':");

        // Виведення всіх предметів після видалення
        foreach (DictionaryEntry entry in subjects)
        {
            Console.WriteLine($"{entry.Key} : {entry.Value}");
        }
    }
```

Рисунок 1 – Програмна реалізація хеш-таблиці

Розуміння принципів роботи хеш-таблиць є важливим складником знань програміста, оскільки ці структури даних дають змогу створювати ефективні алгоритми для роботи з великими наборами даних. Завдяки їм можна оптимізувати продуктивність систем, зменшити час виконання задач та забезпечити швидкий доступ до необхідної інформації [5].

Список використаних джерел

1. Думка Експерта. 2024. URL: https://luka.almedia.com.ua/ukraincyam/de-vikoristovuietsya-kheshuvannya-v-realnomu-zhitti.html (дата звернення 30.11.2024).

2. Чен С. Guru99. 2024. URL: https://www.guru99.com/uk/hash-table-data-structure.html (дата звернення 30.11.2024).

УДК 519.17:004.021

Радченко Р. П., здобувач вищої освіти,
Сеник І. О., асистент кафедри
інформаційних технологій

ГРАФИ ТА АЛГОРИТМИ ПОШУКУ: ТЕОРІЯ, ЗАСТОСУВАННЯ ТА ОПТИМІЗАЦІЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Графи є універсальною структурою даних, яка дає змогу моделювати широкий спектр реальних об’єктів та процесів, як-от соціальні мережі, транспортні системи, ієрархії. Завдяки своїй гнучкості графи використовуються для розв’язання задач, що виникають у різних галузях комп’ютерних наук.

Основу роботи з графами становлять алгоритми пошуку. Серед них найпоширенішими є пошук у ширину (BFS) та пошук у глибину (DFS). Ці алгоритми допомагають обійти всі вершини графу, визначити його зв’язність, знайти шляхи між заданими вершинами та вирішити інші базові задачі.

Для знаходження найкоротших шляхів у графах використовуються більш складні алгоритми, як-от алгоритм Дейкстри та алгоритм Беллмана–Форда. Алгоритм Дейкстри забезпечує ефективне знаходження найкоротших шляхів у графах з невід’ємними вагами, тоді як алгоритм Беллмана–Форда здатен працювати з графами, що містять від’ємні ваги.

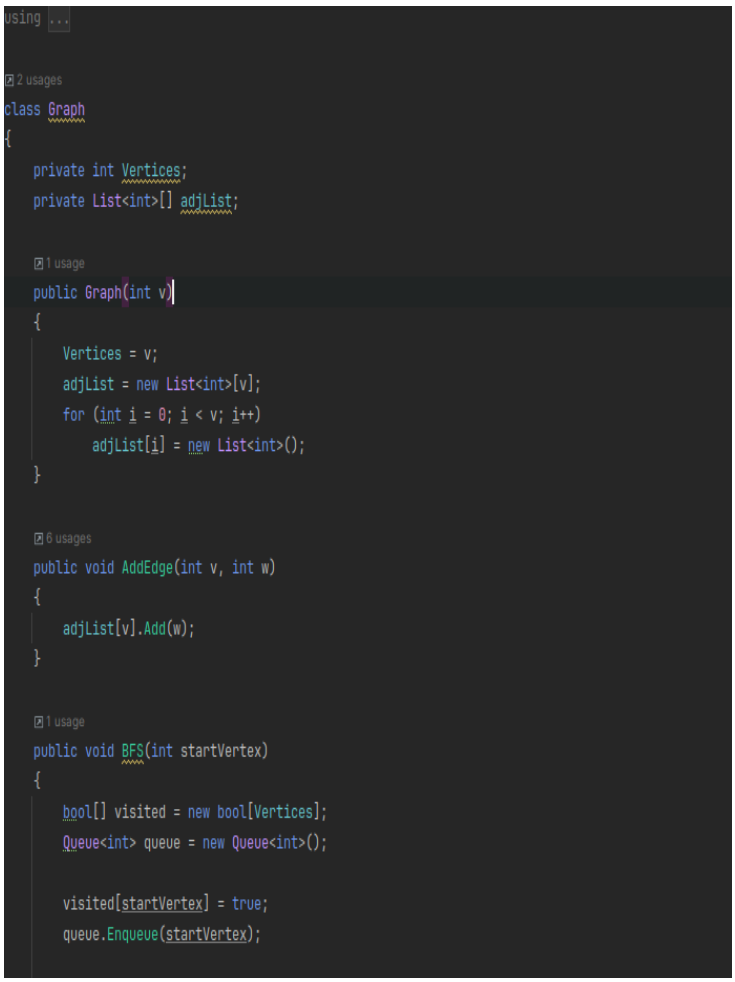
Алгоритми на графах знаходять своє застосування в реальних системах. Наприклад, вони широко використовуються для оптимізації маршрутів у транспортних мережах, де важливо мінімізувати час або вартість переміщення. У соціальних мережах графові алгоритми дають змогу аналізувати зв’язки між користувачами, що допомагає у виявленні впливових осіб або побудові рекомендаційних систем. Важливою задачею є побудова мінімального кістякового дерева. Алгоритми Пріма та Крускала допомагають знайти дерево, яке з’єднує всі вершини графу з мінімальною загальною вагою. Це має практичне значення у проєктуванні мереж, наприклад, електричних чи комунікаційних.

Аналіз складності алгоритмів на графах є ключовим аспектом їх застосування. Залежно від розміру та властивостей графу обирають той чи інший алгоритм, який забезпечить оптимальний результат за прийнятний час.

Програма нижче демонструє базову реалізацію алгоритму BFS для графу, представленого у вигляді списку суміжності. Алгоритм проходить всі вершини графу і виводить порядок їх відвідування (рис. 1 та рис. 2).

Цей код демонструє просту реалізацію алгоритму BFS, який дає змогу проходити всі вершини графу від вказаної вершини. BFS широко використовується в задачах, що потребують знаходження найкоротшого шляху у незваженому графі або перевірки зв’язності графу [4].

Графи та алгоритми пошуку продовжують відігравати ключову роль у комп’ютерних науках і технологіях, забезпечуючи основи для ефективного вирішення складних задач у реальному світі.



```
using ...

2 usages
class Graph
{
    private int Vertices;
    private List<int>[] adjList;

    1 usage
    public Graph(int v)
    {
        Vertices = v;
        adjList = new List<int>[v];
        for (int i = 0; i < v; i++)
            adjList[i] = new List<int>();
    }

    6 usages
    public void AddEdge(int v, int w)
    {
        adjList[v].Add(w);
    }

    1 usage
    public void BFS(int startVertex)
    {
        bool[] visited = new bool[Vertices];
        Queue<int> queue = new Queue<int>();

        visited[startVertex] = true;
        queue.Enqueue(startVertex);
```

Рисунок 1 – Код реалізації алгоритму BFS для графу, представленого у вигляді списку суміжності. Частина 1

```
        while (queue.Count != 0)
        {
            startVertex = queue.Dequeue();
            Console.WriteLine("Відвідано вершину: " + startVertex);

            foreach (int nextVertex in adjList[startVertex])
            {
                if (!visited[nextVertex])
                {
                    visited[nextVertex] = true;
                    queue.Enqueue(nextVertex);
                }
            }
        }
    }
}

class Program
{
    static void Main()
    {
        Graph g = new Graph(v: 5);
        g.AddEdge(v: 0, w: 1);
        g.AddEdge(v: 0, w: 2);
        g.AddEdge(v: 1, w: 2);
        g.AddEdge(v: 2, w: 0);
        g.AddEdge(v: 2, w: 3);
        g.AddEdge(v: 3, w: 3);

        Console.WriteLine("BFS починаючи з вершини 2:");
        g.BFS(startVertex: 2);
    }
```

Рисунок 2 – Код реалізації алгоритму BFS для графу, представленого у вигляді списку суміжності. Частина 2

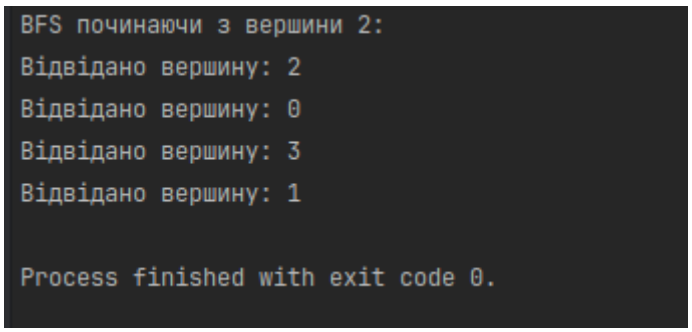


Рисунок 3 – Результат виконання коду алгоритму BFS для графу, представленого у вигляді списку суміжності

Список використаних джерел

1. BestProg. URL: https://www.bestprog.net/uk/2019/09/26/c-queue-general-concepts-ways-to-implement-the-queue-implementing-a-queue-as-a-dynamic-array-ua/ (дата звернення 30.11.2024).

2. GeeksforGeeks. URL: https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/ (дата звернення 30.11.2024).

3. Vseosvita. URL: https://vseosvita.ua/lesson/rol-informatsiinykh-tekhnolohii-u-zhytti-suchasnoi-liudyny-229384.html (дата звернення 30.11.2024).

4. Програмна реалізація та дослідження алгоритмів паралельного швидкого сортування / В. О. Денисюк, Н. А. Потапова, О. В. Зелінська, М. Б. Тарасюк. *Вісник Хмельницького національного університету. Технічні науки.* 2023. № 4. С. 95–105. URL: http://journals.khnu.km.ua/vestnik/?page_id=41

УДК 004.422.5.

Ратушний О. С., здобувач вищої освіти,
Сеник І. О., асистент кафедри
інформаційних технологій

ОСОБЛИВОСТІ ВИКОРИСТАННЯ СТРУКТУРИ ДАНИХ «ГРАФ» У РОЗРОБЦІ ВІДЕОІГОР

Донецький національний університет імені Василя Стуса, м. Вінниця

Граф – це структура даних, що складається з вершин (вузлів) і зв’язків між ними (ребер). Ця структура широко застосовується у комп’ютерних науках для моделювання взаємодій між об’єктами, оскільки вона дає змогу представляти складні системи у вигляді абстрактних моделей. У розробці відеоігор графи відіграють важливу роль, забезпечуючи механізми для навігації, генерації ігрових світів, роботи зі штучним інтелектом та інших ключових функцій [1, 2].

Графи особливо корисні для моделювання ігрових світів. Наприклад, карта гри може бути представлена як граф, де вершини відповідають локаціям, а ребра – шляхам між ними. Алгоритми пошуку найкоротшого шляху, як-от A* або Дейкстри, допомагають персонажам ефективно переміщатися ігровим середовищем, що важливо для створення реалістичних ігрових сценаріїв [3]. Навіть у випадку процедурної генерації рівнів графи використовуються для зв’язування окремих кімнат або областей, створюючи цілісну ігрову карту.

Штучний інтелект у відеоіграх також активно застосовує графи. Логіка NPC часто базується на деревоподібних структурах, які є спеціальними видами графів [4]. Наприклад, дерева прийняття рішень дають змогу персонажам реагувати на дії гравця, а шляхові графи забезпечують ефективне пересування в ігровому середовищі. Це підвищує реалістичність і динаміку взаємодії в грі.

Важливим аспектом використання графів у відеоіграх є їх здатність забезпечувати складні механізми взаємодії. Наприклад, у багатьох іграх графи використовуються для моделювання відносин між персонажами, що допомагає створювати цікаві сюжетні лінії та динамічні взаємодії між гравцями. Ці графи можуть описувати як прямі зв'язки між персонажами, так і складні соціальні структури, включно зі впливами, довірою і конфліктами між ними [4].

Соціальні взаємодії в багатокористувацьких іграх також часто представлені графами. Наприклад, мережа гравців у грі може бути описана у вигляді графу, що дає змогу аналізувати зв'язки між гравцями, їх вплив один на одного або знаходити оптимальні комбінації для командної гри [5]. Такі графи також використовуються для рекомендацій, наприклад, пропонуючи нових друзів або партнерів для гри на основі спільних інтересів.

Графи також знаходять застосування в інших аспектах розробки відеоігор, як-от оптимізація ігрових процесів. Наприклад, для моделювання бойових систем графи можуть бути використані для опису зв'язків між різними тактичними елементами, як-от атаки, оборона і спеціальні навички персонажів. У таких випадках вершини графу можуть представляти різні стани, а ребра – можливі переходи між ними залежно від дій гравця або NPC. Це допомагає моделювати системи взаємодії та прийняття рішень у реальному часі [4].

Однією з основних проблем використання графів у розробці є висока обчислювальна складність, особливо в іграх з великими та динамічними світами. Операції з великими графами можуть бути надто ресурсозатратними, що вимагає застосування спеціалізованих алгоритмів та структур даних для оптимізації. Наприклад, для покращення швидкості обробки даних використовуються методи пошуку в глибину чи ширину, а також оптимізація пам'яті за допомогою більш компактних графів. До того ж застосування структур, як от дерева чи стиснуті графи, може значно знизити навантаження на систему [3]. Однак ці труднощі компенсуються широкими можливостями графів у вирішенні складних завдань, а саме моделювання світу гри, маршрутизація, або управління соціальними взаємодіями. Отже, хоча обчислювальна складність є важливою проблемою, вона не зменшує цінність графів у створенні складних ігрових механік.

Отже, графи є основним інструментом у розробці відеоігор, який допомагає створювати складні та захопливі ігрові світи. Їх правильне використання забезпечує реалізацію реалістичних механік, що є ключовим аспектом сучасних відеоігор.

Список використаних джерел

1. Graphs in Game Theory. Medium. URL: https://medium.com/@jshreyas12/graphs-in-game-theory-8c6c09fa5d45 (дата звернення: 02.12.2024).

2. Data Structures for Game Developers. Medium. URL: https://medium.com/supercent-blog/data-structures-for-game-developers-2d2e0adcc931 (дата звернення: 02.12.2024).

3. 10 Data Structures and Algorithms for Game Developers. Analytics Insight. URL: https://www.analyticsinsight.net/latest-news/10-data-structures-and-algorithms-for-game-developers (дата звернення: 02.12.2024).

4. AI Transforming the Role of NPCs in Modern Games. GameCloud. URL: https://gamecloud-ltd.com/ai-transforming-the-role-of-npcs-in-modern-games/ (дата звернення: 02.12.2024).

5. Video Game Dialogues and Graph Theory. URL: https://philipphagenlocher.de/post/video-game-dialogues-and-graph-theory/ (дата звернення: 02.12.2024).

УДК 004.8

Родюк К. О., здобувачка вищої освіти,
Луценко А. В., доктор філософії
з математики, старший викладач,
в. о. завідувача кафедри прикладної
математики та кібербезпеки

ЗАСТОСУВАННЯ МАТЕМАТИКИ У ГЛИБИННОМУ НАВЧАННІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Анотація. У роботі йдеться про застосування математики у глибинному навчанні. Особливу увагу приділено важливості розділів математики, як-от лінійна алгебра, теорія ймовірностей, теорія диференціальних рівнянь та математичний аналіз, в контексті навчання нейронних мереж. На основі цього підкреслено важливий внесок математики у глибинне навчання.

Ключові слова: математика, глибинний аналіз, лінійна алгебра, теорія ймовірності, диференціальні рівняння, математичний аналіз.

Вступ. У сучасному світі глибинне навчання відіграє вирішальну роль у розвитку технологій, які мають значний вплив на різні сфери життя. Використання нейронних мереж у глибинному навчанні дає змогу створювати системи штучного інтелекту, які здатні розпізнавати образи, аналізувати дані та приймати складні рішення з високою точністю. Це має величезне значення для медичної діагностики, автоматизації виробництва, розвитку автономного транспорту та багатьох інших галузей, де точність і ефективність є критично важливими.

Математика є основою для розуміння і розвитку глибинного навчання. Базові математичні концепції, як-от лінійна алгебра, теорія ймовірностей, диференціальні рівняння та математичний аналіз, забезпечують теоретичний фундамент, необхідний для побудови та оптимізації нейронних мереж. Без глибокого розуміння цих дисциплін важко створити моделі, які ефективно навчалися б на великих обсягах даних та демонстрували високу продуктивність. Тому метою цієї роботи є дослідження необхідних математичних знань для глибинного навчання та їх практичне застосування у створенні сучасних алгоритмів машинного навчання.

Основний текст. Глибинне навчання – це підгалузь машинного навчання, яка застосовує нейронні мережі з численними шарами для обробки великих обсягів інформації. Щоб повноцінно зрозуміти глибинне навчання, спочатку

необхідно ознайомитися з основами машинного навчання, а також мати ґрунтовні знання з математики [1, 2, 3].

Алгоритм машинного навчання – це алгоритм, який має здатність навчатися на основі даних. З інженерної перспективи машинне навчання дає змогу розв'язувати задачі, які не можуть бути розв'язані за допомогою програм, створених людьми. З наукового погляду машинне навчання є цікавим, оскільки його вивчення допомагає зрозуміти деякі принципи, що лежать в основі розумної поведінки.

Навчання – це здатність виконувати завдання. Завдяки машинному навчанню можна розв'язувати різноманітні типи задач. Найпоширенішими завданнями машинного навчання є: класифікація (програму просять визначити, до якої з категорій належить певний вхід, для розв'язання зазвичай пропонується створити функцію); класифікація з відсутніми вхідними даними (алгоритм повинен визначити відображення однієї функції від векторного вводу до категоріального виходу); транскрипція (необхідно спостерігати за неструктурованим представленням певного типу даних і транскрибувати їх у вигляді дискретної текстової форми); переклад (комп'ютерна програма повинна перетворити вже сформовані вхідні дані з послідовності символів певною мовою на послідовність символів іншою мовою).

Основними математичними концепціями у глибинному навчанні є:

Лінійна алгебра. Однією з фундаментальних дисциплін для розуміння глибинного аналізу є лінійна алгебра. Вона використовується для обробки даних, для дій над скалярами, векторами, матрицями й тензорами. Також використовується для розв'язання задач з великою кількістю даних.

Множення матриць є однією з основних операцій для обчислення вхідних значень нейронів. Вектори використовуються для перетворення зображення в одновимірний вектор пікселів. Тензори використовують для обробки даних у складніших моделях.

Теорія ймовірності. Теорія ймовірності так само, як і лінійна алгебра, є однією із фундаментальних дисциплін для глибинного аналізу. Вона допомагає створити модель, яка прогнозує ймовірність події або її результат. Теорія ймовірностей також може використовуватись для теоретичного аналізу систем штучного інтелекту. До того ж вона є основним знаряддям для оцінки похибок.

Теорія диференціальних рівнянь. Диференціальні рівняння мають вплив на глибинне навчання. Насамперед вони впливають на моделювання динамічних процесів, а також використовуються для роботи з даними, які мають часову залежність. Диференціальні рівняння допомагають аналізувати стабільність моделі.

Математичний аналіз. Математичний аналіз забезпечує теоретичний базис для оптимізації та оцінки похідних, а також забезпечує основу для побудови алгоритмів та вивчення функцій втрат.

Основним методом для оптимізації в глибинному навчанні є градієнтний спуск. Його мета – мінімізувати функцію втрат. Також використовуються інтеграли для обчислення значень змінних.

Висновки. Глибинне навчання є складною і потужною галуззю машинного навчання, яка використовує нейронні мережі для обробки великих обсягів даних.

Для повного розуміння цієї технології необхідно мати ґрунтовні знання основ машинного навчання та деяких розділів математики, як-от лінійна алгебра, теорія ймовірностей, теорія диференціальних рівнянь та математичний аналіз.

Машинне навчання дає змогу вирішувати різноманітні задачі, які не можуть бути розв'язані традиційними програмами, а також допомагає зрозуміти принципи розумної поведінки.

Основні задачі машинного навчання включають класифікацію, транскрипцію та переклад, а методи оптимізації, як-от градієнтний спуск, відіграють ключову роль у досягненні успішних результатів.

Отже, математика відіграє вважливу роль у глибинному навчанні, а розвиток нейронних мереж сприяє прогресу в майбутньому.

Список використаних джерел

1. Neural Ordinary Differential Equations. arXiv / R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud. 2018. 12 p.

2. Michelucci U. Fundamental Mathematical Concepts for Machine Learning in Science. Dubendorf: Springer, 2024. 249 p.

3. Goodfellow I., Bengio Y., Courvill A. Deep Learning. Cambridge, Massachusetts: MIT Press, 2016. 800 p.

УДК 004.422.5

Слободяник Р. Р., здобувач вищої освіти,
Поремський Ю. В., канд. техн. наук,
старший викладач кафедри інформаційних технологій

ОСОБЛИВОСТІ ВИКОРИСТАННЯ АЛГОРИТМІВ У КІБЕРБЕЗПЕЦІ

Донецький національний університет імені Василя Стуса, м. Вінниця

У світі сучасних інформаційних технологій алгоритми відіграють вирішальну роль у забезпеченні кібербезпеки. Вони застосовуються для захисту систем, мереж і даних від несанкціонованого доступу, модифікації чи втрати. Одним із головних завдань кібербезпеки є забезпечення конфіденційності, цілісності та доступності інформації, що досягається через використання алгоритмів шифрування, хешування, цифрових підписів, а також методів машинного та глибинного навчання [1].

Алгоритми шифрування є основою кіберзахисту. Вони дають змогу перетворювати відкриті дані у зашифровану форму, недоступну для сторонніх. Наприклад, Advanced Encryption Standard (AES) використовується у багатьох сучасних системах для захисту конфіденційних даних. Алгоритми типу RSA

забезпечують ефективний захист завдяки використанню пар ключів – приватного та публічного [1].

Алгоритми хешування (SHA-256, MD5) забезпечують перевірку цілісності даних і безпечне зберігання паролів. Наприклад, під час введення користувачем пароля система порівнює збережений у базі хеш значення з отриманим після хешування введеного пароля. Це виключає ризик зберігання відкритих паролів [1].

Натомість цифрові підписи, реалізовані за допомогою алгоритмів, як-от ECDSA, забезпечують автентичність електронних документів та повідомлень. Вони допомагають ідентифікувати підписанта та виявляти зміни в документі, якщо вони відбулися [1].

Крім традиційних алгоритмів, активно розвиваються методи машинного навчання (ML) та глибинного навчання (DL). Ці технології дають змогу ефективно виявляти загрози шляхом аналізу великих обсягів даних. Наприклад, Convolutional Neural Networks (CNN) аналізують мережевий трафік для виявлення аномалій, а Graph Neural Networks (GNN) моделюють складні зв’язки у кібербезпекових даних [2, 3].

Моделі на основі Federated Learning (FL) дають змогу забезпечувати конфіденційність даних під час навчання моделей машинного навчання, оскільки дані залишаються на локальних пристроях. Такий підхід застосовується у фінансових системах для виявлення шахрайства [2].

Сучасні розробки також включають технологію Adversarial Learning, яка забезпечує стійкість систем до атак з використанням модифікованих даних. Ця технологія дає змогу виявляти навіть складні форми шкідливих дій [2].

Розуміння ролі алгоритмів у кібербезпеці є ключовим для створення стійких систем захисту. Вони забезпечують основу для побудови ефективних механізмів захисту від загроз та створення безпечного цифрового простору.

Список використаних джерел

1. Що таке алгоритм. Терміни та визначення кібербезпеки. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/algorithm (дата звернення 30.11.2024).

2. Machine Learning Algorithms for Detecting and Preventing Cyber Threats. URL: https://www.oxjournal.org/machine-learning-algorithms-for-detecting-and-preventing-cyber-threats/ (дата звернення 30.11.2024).

3. Deep Learning Algorithms for Cybersecurity Applications: A Technological and Status Review. URL: https://www.sciencedirect.com/science/article/abs/pii/S157401372030417 (дата звернення 30.11.2024).

УДК 004.045:621.396.967.2(043.2)

Труханська В. О., здобувачка вищої освіти,
Ніколюк П. К., д-р фіз.-мат. наук,
професор, професор кафедри
інформаційних технологій

ОБРОБКА СИГНАЛІВ ДЛЯ ВИЯВЛЕННЯ ПОВІТРЯНИХ ЦІЛЕЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

Виявлення повітряних цілей, як-от дрони, літаки або ракети, є одним з актуальних завдань сучасної радіолокації та акустичного моніторингу. У міру розвитку технологій зростає потреба у використанні автоматизованих систем для ідентифікації цілей, що базуються на аналізі акустичних сигналів. Такі системи дають змогу виконувати моніторинг у реальному часі, що знижує ризики невиявлення небезпечних об’єктів [1]. Завдяки використанню методів цифрової обробки сигналів (ЦОС) завдання виявлення цілей може бути вирішене з високою точністю навіть у складних умовах, зокрема за наявності шуму або перешкод.

У цій роботі представлено методологію аналізу акустичних сигналів для виявлення характерних частот, пов’язаних з повітряними цілями. Реалізований алгоритм базується на швидкому перетворенні Фур’є (FFT), методах пошуку піків і порівнянні виявлених частот із наперед заданим набором цільових характеристик [2].

Процес виявлення повітряних цілей акустичними методами включає кілька ключових етапів. Спершу йде попередня обробка сигналу, де здійснюється завантаження аудіофайла та приведення сигналу до моноформату. Далі здійснюється спектральний аналіз: за допомогою швидкого перетворення Фур’є (FFT) отримується частотний спектр сигналу, що допомагає перейти від часової області до частотної [1]. Оскільки більшість інформаційних частот містяться у позитивному спектрі, аналіз обмежується цим діапазоном. Результатом є амплітудний спектр сигналу. Наступне – це виявлення характерних частот, використовується алгоритм пошуку піків для ідентифікації частот із високою амплітудою. Ці частоти порівнюються з наперед заданим списком характерних частот, що відповідають потенційним цілям (наприклад, дронам, літакам тощо) [3]. Алгоритм допускає певний поріг амплітуди для уникнення помилкових виявлень через шум. Останнім кроком є візуалізація результатів, що представляються у вигляді графіків, які демонструють сигнал у часовій області та його спектр у частотній. На спектрі також вказуються виявлені частоти цілей.

Розроблений алгоритм реалізовано мовою Python із використанням бібліотек numpy, matplotlib та scipy. Код представляє процес обробки сигналу, попередньо завантаживши аудіофайл, і графічний вивід представлення результатів.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import wavfile
from scipy.signal import find_peaks
# === Параметри для виявлення цілей ===
```

```
# Частоти, характерні для повітряних цілей (дрони, літаки, ракети)
target_frequencies = [150, 300, 450, 800, 1200]  # Цілі (можна розширити)
sampling_rate = 44100  # Типова частота дискретизації для аудіофайлів (Hz)
threshold = 10  # Поріг для виявлення сигналу
# === Завантаження реального аудіофайла ===
def load_audio(file_path):
    rate, data = wavfile.read(file_path)
    return rate, data
# === Аналіз сигналу ===
def analyze_signal(signal, rate):
    fft_spectrum = np.fft.fft(signal)
    freq = np.fft.fftfreq(len(fft_spectrum), d=1 / rate)
    # Повертаємо лише позитивні частоти
    positive_freqs = freq[freq > 0]
    positive_spectrum = np.abs(fft_spectrum[freq > 0])
    return positive_freqs, positive_spectrum
# === Виявлення частот цілей ===
def detect_targets(positive_freqs, positive_spectrum, threshold=5):
    # Знаходимо піки у спектрі
    peaks, _ = find_peaks(positive_spectrum, height=threshold)
    detected_frequencies = positive_freqs[peaks]
    return detected_frequencies
# === Основна програма ===
def main():
    file_path = "drone_sound.wav"  # Шлях до мого аудіофайла
    print(f"Завантаження аудіофайлу: {file_path}")
    # Завантаження аудіофайла
    rate, data = load_audio(file_path)
    # Якщо файл стерео, перетворюємо в моно
    if len(data.shape) > 1:
        data = data.mean(axis=1)
    print("Аналіз сигналу...")
    positive_freqs, positive_spectrum = analyze_signal(data, rate)
    print("Виявлення цілей...")
    detected_frequencies = detect_targets(positive_freqs, positive_spectrum, threshold)
    # Виведення результатів
    print("Виявлені частоти цілей (Гц):", detected_frequencies)
    for freq in detected_frequencies:
        if freq in target_frequencies:
            print(f"Ціль виявлена на частоті {freq} Гц!")
        else:
            print(f"Виявлено невідомий сигнал на частоті {freq} Гц.")
    # Побудова графіків
    plt.figure(figsize=(10, 6))
    # Сигнал у часі
    plt.subplot(2, 1, 1)
    time = np.arange(0, len(data)) / rate
    plt.plot(time, data, label="Сигнал")
    plt.title("Сигнал у часі")
    plt.xlabel("Час (с)")
    plt.ylabel("Амплітуда")
    plt.legend()
    # Частотний спектр
```

```
    plt.subplot(2, 1, 2)
    plt.plot(positive_freqs, positive_spectrum, label="Спектр сигналу")
    plt.scatter(
        detected_frequencies,
        positive_spectrum[[np.where(positive_freqs == f)[0][0] for f in detected_frequencies]],
        color="red", label="Виявлені цілі"
)
    plt.title("Частотний спектр сигналу")
    plt.xlabel("Частота (Гц)")
    plt.ylabel("Амплітуда")
    plt.legend()
    plt.tight_layout()
    plt.show()
# === Запуск програми ===
if __name__ == "__main__":
```

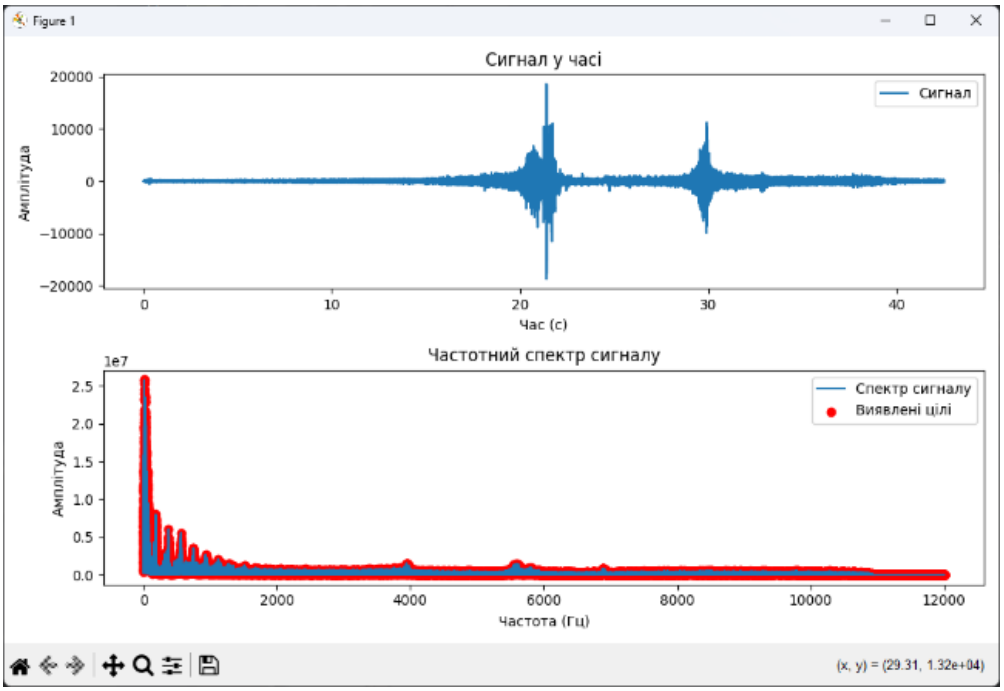


Рисунок 1 – Графіки представлення результатів програми

Алгоритм успішно ідентифікує частоти, відповідні цільовим характеристикам. Візуалізація у вигляді графіків демонструє точність аналізу: сигнал у часовій області і спектральний склад сигналу дають змогу оцінити як параметри сигналу, так і ефективність виявлення.

Висновки. Розроблений метод обробки сигналів довів свою практичну цінність у завданнях виявлення повітряних цілей. Його гнучкість забезпечується модульністю реалізації та можливістю розширення функціоналу. Подальше вдосконалення може включати адаптацію алгоритму до шумових середовищ, підвищення чутливості виявлення шляхом оптимізації порогів, а також інтеграцію методів класифікації цілей.

Список використаних джерел

1. Методика прогнозу тенденцій розвитку засобів військової радіолокації виявлення повітряних цілей. DOI: 10.30748/nitps.2021.43.15 (дата звернення 27.11.2024).
2. Verbytskyi I. V. Швидке перетворення Фур'є модульованих сигналів, представлених рядом Фур'є двох змінних. *Вісник Національного технічного університету «ХПІ». Серія: Нові рішення у сучасних технологіях*. 2018 № 16(1292). С. 102–106. DOI: 10.20998/2413-4295.2018.16.15 (дата звернення 26.11.2024).
3. The Creation of a Hidden Radar Field for the Detection of Small Aerial Objects due to the Use of Signals from Telecommunication Systems / Khudov et. al. *International Journal of Applied Engineering and Technology*. 2023. № 5(3). P. 9–17.

УДК 517.9:004.43(043.2)

Хріпко О. І., здобувач вищої освіти,
Фриз І. В., канд. фіз.-мат. наук, старший викладач кафедри інформаційних технологій

РОЗВ'ЯЗУВАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ НА МОВІ PYTHON

Донецький національний університет імені Василя Стуса, м. Вінниця

Сьогодні існують різноманітні технології, за допомогою яких обробка значних об'ємів даних проходить практично миттєво. Важливу роль відіграє математичний апарат, який є одним із базових інстументів для дослідження і моделювання складних систем. Зокрема, потужним математичним інструментарієм є диференціальні рівняння, які використовуються для математичного моделювання динамічних систем, під час дослідження і прогнозування реальних фізичних, біологічних, екологічних, соціально-економічних та інших процесів і явищ.

Ефективне розв'язування та дослідження диференціальних рівнянь і їх систем стало можливим завдяки різним інтерфейсам та бібліотекам, які забезпечують засобами для розв'язування диференціальних рівнянь, зокрема і за допомогою потужних модулів на базі Python. Отже, метою є вивчення можливостей для розв'язування та графічної візуалізації диференціальних рівнянь, зокрема за допомогою функції odeint бібліотеки SciPy і бібліотеки MatPlotLib.

Бібліотека SciPy, зокрема і її функція *odeint,* – це бібліотека, що надає користувачу базу для чисельного розв'язування диференціальних рівнянь [1], а бібліотека MatPlotLib містить інструментарій для візуалізації отриманих розв'язків. Інтерфейс odeint є ефективним для розв'язування звичайних диференціальних рівнянь, однак не завжди може бути застосованим для інших типів рівнянь.

На початку роботи під час розв'язування диференціальних рівнянь потрібно встановити необхідні бібліотеки: SciPy, NumPy, MatPlotLib. Розглянемо використання цих бібліотек [2] на прикладі диференціального рівняння із заданими початковими умовами для декількох значень параметра k:

$$\frac{dy(t)}{dt} = -k * y(t) \qquad (1)$$

Для чисельного розв’язання потрібно задати початкові умови. Відповідний код має вигляд:

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

def model(y,t,k):
    dydt = -k * y
    return dydt

y0 = 5
t = np.linspace(0,20)

k = 0.1
y1 = odeint(model,y0,t,args=(k,))
k = 0.2
y2 = odeint(model,y0,t,args=(k,))
k = 0.5
y3 = odeint(model,y0,t,args=(k,))

plt.plot(t,y1,'r-',linewidth=2,label='k=0.1')
plt.plot(t,y2,'b--',linewidth=2,label='k=0.2')
plt.plot(t,y3,'g:',linewidth=2,label='k=0.5')
plt.xlabel('time')
plt.ylabel('y(t)')
plt.legend()
plt.show()
```

Графічний результат зображений на рис. 1, зокрема отримані інтегральні криві показують, як може змінюватися поведінка $y(t)$ для різних значень коефіцієнта k.

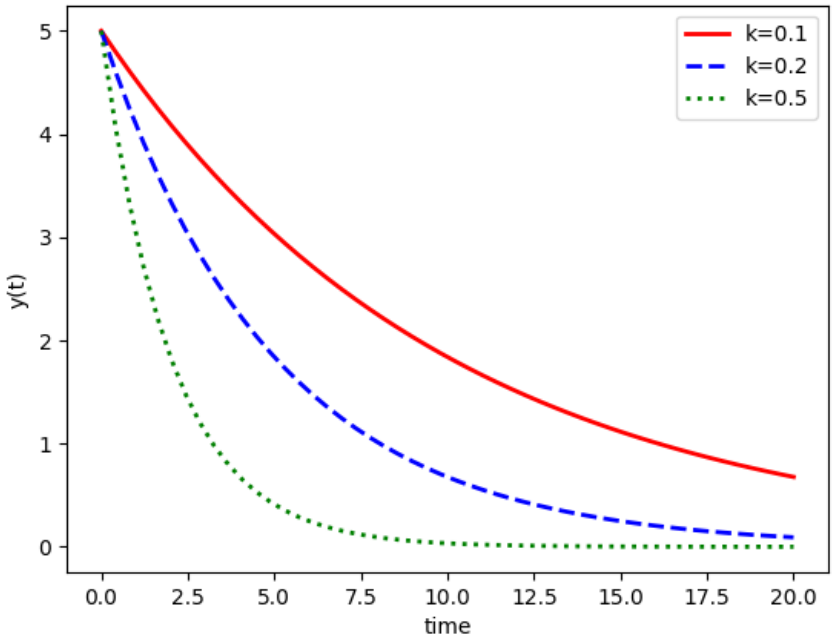


Рисунок 1 – Графічна візуалізація розв’язків диференціального рівняння

Варто зазначити, що SciPy містить ще функцію *solve_ivp,* яка є більш новим інтерфейсом. Для кодів, які використовують останню версію бібліотеки SciPy, рекомендується використовувати solve_ivp, оскільки вона надає ширшу базу для розв'язання диференціальних рівнянь та їх систем. Solve_ivp підтримує декілька чисельних методів розв'язування диференціальних рівнянь (методи Рунге–Кутта, BDF-метод тощо) і надає можливість більш гнучкого налаштування цього процесу. До того ж є можливість вручну обрати метод інтегрування, наприклад, RK23, RK45, BDF, LSODA тощо, водночас *odeint* містить адаптивний метод LSODA. Для детальнішого ознайомлення з наведеними функціями та їх аргументами можна звернутися до [1] і [3].

Отже, розв'язання диференціальних рівнянь за допомогою Python є не тільки ефективним, але й швидким за допомогою можливостей, які надають бібліотеки SciPy та MatPlotLib.

Список використаних джерел

1. Integration and ODEs (scipy.integrate) – SciPy v1.14.1 Manual. *Numpy and Scipy Documentation – Numpy and Scipy documentation*. URL: https://docs.scipy.org/doc/scipy/reference/integrate.html (date of access: 01.12.2024).

2. Solve Differential Equations with ODEINT. *APMonitor Optimization Suite*. URL: https://apmonitor.com/pdc/index.php/Main/SolveDifferentialEquations (date of access: 01.12.2024).

3. Müller-Komorowska D. Differential Equations with SciPy – odeint or solve_ivp. *Simulation-Based*. URL: https://simulationbased.com/2021/02/16/differential-equations-with-scipy-odeint-or-solve_ivp/ (date of access: 01.12.2024).

УДК 519.22/.25:004.415]:332.624(043.2)

Чемес В. С., здобувач вищої освіти,
Хмелівський Ю. С., асистент кафедри
інформаційних технологій

ЗАСТОСУВАННЯ ЛІНІЙНОЇ РЕГРЕСІЇ ДЛЯ ОЦІНКИ ЦІНИ НЕРУХОМОСТІ ЗА РІЗНИМИ ПАРАМЕТРАМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Ринок нерухомості є однією з найбільш активних і важливих частин економіки. Визначення справедливої ціни об'єкта є однією з головних проблем для інвесторів, покупців і продавців нерухомості. Ціна нерухомості залежить від багатьох факторів, зокрема місцезнаходження, площі, стану будівлі, кількості кімнат, наявності додаткових зручностей тощо. Отже, передбачення ціни є складним завданням, і щоб отримати точну оцінку, необхідно використовувати математичні моделі.

Лінійна регресія є одним із найбільш поширених методів вирішення цієї проблеми. Лінійна регресія дає змогу оцінити вплив різних елементів на ціну нерухомості та створити модель, яка може прогнозувати ціну на основі значень цих елементів. Це особливо важливо на ринку нерухомості, оскільки зміна одного фактора може значно вплинути на кінцеву вартість нерухомості.

Однак оцінка ціни нерухомості залежить від кількох факторів, і застосування лінійної регресії з однією змінною не є доцільним, оскільки це є недостатнім для точного прогнозування. Тому для більш точних і реалістичних результатів доцільно використовувати множинну лінійну регресію, яка допомагає враховувати кілька незалежних змінних одночасно.

У такій моделі залежна змінна (ціна нерухомості) оцінюється на основі комбінації кількох факторів, що значно підвищує точність прогнозів та дає змогу врахувати складні взаємозв'язки між різними характеристиками нерухомості [1].

Множинна лінійна регресія є розширенням лінійної регресії, де для прогнозування залежної змінної використовується більше ніж одна незалежна змінна. Це один із основних методів статистичного навчання, який допомагає оцінити вплив кількох факторів на результат і побудувати математичну модель для прогнозування значення залежної змінної.

Множинна лінійна регресія дає змогу створювати рівняння, яке описує лінійний зв'язок між кількома незалежними змінними (предикторами) та залежною змінною. Математичне представлення моделі множинної регресії виглядає так:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + e,$$

де y – залежна змінна (наприклад, ціна нерухомості);

$x_1, x_2, \ldots, x_n$ – незалежні змінні (наприклад, площа, кількість кімнат, район та ін.);

β_0 – вільний член (інтерцепт);

$\beta_1, \beta_2, \ldots \beta_n$ – коефіцієнти регресії, які вказують на вплив кожної незалежної змінної на залежну;

e – випадкова похибка (залишкове значення) [1, 2].

Мета множинної лінійної регресії – знайти такі значення коефіцієнтів $\beta_1, \beta_2, \ldots \beta_n$, які мінімізують похибку моделі. Зазвичай для цього використовують метод найменших квадратів, який мінімізує суму квадратів різниць між прогнозованими і реальними значеннями залежної змінної. Процес виглядає так:

$$min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

де y_i – реальні значення залежної змінної;

$\hat{y}_i$ – прогнозовані значення залежної змінної на основі моделі.

Мета дослідження – побудувати модель множинної лінійної регресії для прогнозування ціни нерухомості (y) на основі різних характеристик квартири. У якості незалежних змінних обрано площу квартири (x_1), кількість кімнат (x_2), відстань до центру міста (x_3), наявність паркінгу (x_4) [4].

Необхідно знайти коефіцієнти моделі, які допомагають найкраще описати вплив зазначених факторів на ціну квартири.

Для побудови моделі використовується вибірка даних про квартири (табл. 1).

Таблиця 1 – Вибірка даних про квартири

Площа, м²	**Кількість кімнат**	**Відстань до центру, км**	**Паркінг**	**Ціна, $**
50	2	5	0	100 000
70	3	3	1	150 000
45	1	8	0	80 000
120	4	2	1	250 000
90	3	4	1	200 000

З огляду на умову завдання множинна лінійна регресія описується рівнянням:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + e.$$

Для оцінки параметрів регресії застосовуємо метод найменших квадратів. Модель у матричній формі:

$$\mathrm{y} = \mathrm{X}\beta + \mathrm{e},$$

де y – вектор залежної змінної;
X – матриця незалежних змінних;
β – вектор коефіцієнтів;
e – вектор похибок.
Для оцінки β використовується формула:

$$\beta = (\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T\mathrm{y}\ [3].$$

Використовуючи дані таблиці, формуємо матриці:

$$\mathrm{X} = \begin{bmatrix} 1 & 50 & 2 & 5 & 0 \\ 1 & 70 & 3 & 3 & 1 \\ 1 & 45 & 1 & 8 & 0 \\ 1 & 120 & 4 & 2 & 1 \\ 1 & 90 & 3 & 4 & 1 \end{bmatrix}, \qquad \mathrm{y} = \begin{bmatrix} 100\,000 \\ 150\,000 \\ 80\,000 \\ 250\,000 \\ 200\,000 \end{bmatrix}$$

Обчисливши матриці, отримуємо:

- β_0 = –1 210 000 – базова ціна;
- β_1 = –4 000 – кожен додатковий м² зменшує ціну на $4 000;
- β_2 = 430 000 – кожна додаткова кімната додає $430 000 до ціни;
- β_3 = 130 000 – кожен кілометр віддаленості збільшує ціну на $130 000;
- β_4 = –40 000 – наявність паркінгу зменшує ціну на $40 000.

Після визначення коефіцієнтів регресії отримано математичну модель, яка описує залежність ціни нерухомості від основних характеристик квартири:

$$y = -1\,210\,000 - 4\,000x_1 + 430\,000x_2 + 130\,000x_3 - 40\,000x_4.$$

Отримана модель дає змогу кількісно оцінити вплив основних факторів на формування ціни нерухомості, як-от площа, кількість кімнат, віддаленість від центру та наявність паркінгу. Вона також забезпечує можливість прогнозувати вартість житла на основі цих параметрів, що є корисним для покупців, продавців та рієлторів.

Висновки. У цій роботі було розглянуто застосування множинної лінійної регресії для прогнозування ціни нерухомості, що допомагає враховувати вплив кількох факторів на результат. Множинна лінійна регресія є потужним інструментом статистичного навчання, який дає змогу оцінити взаємозв'язок між кількома незалежними змінними та залежною змінною.

Прогнозування за допомогою цієї моделі є корисним інструментом для покупців, продавців і рієлторів, оскільки допомагає приймати обґрунтовані рішення. До того ж модель має потенціал для подальших удосконалень, зокрема додавання нових факторів, що можуть впливати на ціни на ринку нерухомості.

Список використаних джерел

1. Kanade V. What Is Linear Regression? Types, Equation, Examples, and Best Practices for 2022. Spiceworks. 2023. URL: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/ (дата звернення 30.11.2024).

2. Geeks for Geegks. Linear Regression in Machine learning. 2023. URL: https://www.geeksforgeeks.org/ml-linear-regression/ (дата звернення 30.11.2024).

3. Wikipedia. General linear model. URL: https://en.wikipedia.org/wiki/General_linear_model (дата звернення 30.11.2024).

4. Yasser M. Housing Prices Dataset. Kaggle. 2022. URL: https://www.kaggle.com/datasets/yasserh/housing-prices-dataset (дата звернення 30.11.2024).

УДК 004.422.5

Черніхевич О. В., здобувач вищої освіти,
Поремський Ю. В., канд. техн. наук,
старший викладач кафедри інформаційних технологій

АЛГОРИТМИ ДИНАМІЧНОГО ПРОГРАМУВАННЯ В ЗАДАЧАХ ОПТИМАЛЬНОГО ВИБОРУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Динамічне програмування (Dynamic Programming, DP) – це метод оптимізації, що використовується для вирішення складних задач шляхом розбиття їх на простіші підзадачі. Динамічне програмування ефективно використовується для задач, які можна розділити на перекриваючі підзадачі з оптимальною структурою підзадач.

1. Мемоізація (зверху вниз): рекурсивний підхід, за якого результати підзадач зберігаються в пам'яті для уникнення повторних обчислень.

2. Табуляція (знизу вверх): ітеративний підхід, за якого результати підзадач обчислюються і зберігаються в таблиці (зазвичай масиві) і використовуються для обчислення кінцевого результату [1].

Метод динамічного програмування (ДП) є одним з основних інструментів оптимізації та розв'язання задач в інформатиці, економіці, біології та інших галузях. Інакше кажучи, динамічне програмування – це метод розв'язання задач, що ґрунтується на розбитті складної задачі на безліч дрібніших.

Важливо враховувати такі моменти:

1. Для ефективного застосування цього підходу необхідно запам’ятовувати рішення підзадач.

2. Підзадачі мають загальну структуру, що дає змогу використовувати однорідний спосіб їх розв’язання, замість того, щоб кожну вирішувати окремо різними алгоритмами [2].

Динамічне програмування зазвичай, але не завжди, використовується для вирішення задач оптимізації, схожих на жадібні алгоритми. На відміну від жадібних алгоритмів, які вимагають, щоб властивість жадібного вибору була дійсною, динамічне програмування працює над низкою проблем, у яких локально оптимальний вибір не дає глобально оптимальних результатів [3].

На практиці рішення задачі за допомогою динамічного програмування включає в себе дві основні частини: налаштування динамічного програмування і подальше виконання обчислень. Налаштування динамічного програмування зазвичай вимагає 5 кроків:

1. Знайти «матричну» параметризацію задачі. Визначте кількість розмірів (змінних).

2. Переконайтеся, що підзадачний простір є поліномом (не експоненціальним). Зверніть увагу, що якщо використовується невелика частина підзадач, то мемоізація може бути кращою; аналогічно, якщо повторне використання підзадач не є великим, динамічне програмування може бути не найкращим рішенням проблеми.

3. Визначте ефективний поперечний порядок: підзадачі повинні бути готові (вирішені), коли вони потрібні, тому порядок обчислень має значення.

4. Визначте рекурсивну формулу: більша задача зазвичай вирішується як функція її підчастин.

5. Запам’ятайте вибір: зазвичай рекурсивна формула передбачає крок мінімізації або максимізації. Навіть більше, часто потрібне подання для зберігання поперечних покажчиків, і подання повинно бути поліномом [3].

Список використаних джерел

1. Focminded. Метод динамічного програмування: ключові аспекти та застосування. 2023. URL: https://foxminded.ua/metod-dynamichnoho-prohramuvannia/

СЕКЦІЯ 2
ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ТА МЕРЕЖІ

УДК 004.021

Зигарь Д. Д., здобувач вищої освіти,
Ніколюк П. К., д-р фіз.-мат. наук,
професор, професор кафедри
інформаційних технологій

РОЗРОБЛЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЇ ЦІЛЕЙ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні технології штучного інтелекту (ШІ) мають значний вплив на розробку систем розпізнавання та класифікації цілей у різних секторах, особливо в оборонній промисловості, безпеці та автономних транспортних засобах [1]. За допомогою штучного інтелекту ми можемо автоматизувати процес виявлення, ідентифікації та класифікації об’єктів, а також підвищити точність і швидкість прийняття рішень [2].

Основа системи розпізнавання і класифікації цілей. Системи розпізнавання та класифікації цілей призначені для автоматичного виявлення об’єктів у навколишньому середовищі, визначення їх характеристик та віднесення їх до певних категорій. Традиційно такі системи базувались на алгоритмах обробки сигналів та зображень, які вимагали ручного налаштування параметрів та обмежувалися змінними середовищами [3].

Впровадження штучного інтелекту. Інтеграція штучного інтелекту, зокрема методів машинного навчання та глибоких нейронних мереж, значно покращила функціональність системи розпізнавання. Машинне навчання забезпечує здатність системи вчитися на великих обсягах даних, виявляти складні закономірності і адаптуватися до нових умов. Глибокі нейронні мережі, особливо згорткові нейронні мережі (Cnn), ефективно обробляють візуальні дані, які мають вирішальне значення для розпізнавання об’єктів на зображеннях та відео.

Етапи розробки системи:

1. Збір і підготовка даних: необхідно зібрати велику кількість зображень або відео з різними типами цілей, з різними умовами освітлення, кутами і фоном. Це означає, що кожне зображення повинно містити інформацію про місцезнаходження та клас об’єкта [4].

2. Вибір та конфігурація моделі: архітектура нейронної мережі вибирається на основі деталей завдання. Для завдань розпізнавання об’єктів часто використовуються моделі, як-от YOLO (який ви бачите лише один раз) та швидший R-CNN, щоб збалансувати точність і швидкість.

3. Навчання моделі: модель навчається на основі підготовлених даних та оптимізує параметри, щоб мінімізувати помилки в класифікації та локалізації об’єктів. Процес навчання може вимагати значних обчислювальних ресурсів і часу [4].

4. Тестування та валідація: після навчання модель тестується на дані, які не використовувалися під час навчання, щоб оцінити її здатність узагальнювати та правильно класифікувати нові об’єкти [5].

5. Впровадження та оптимізація: навчена модель інтегрована в цільову систему й оптимізована для роботи в режимі реального часу та доступних апаратних ресурсів [6].

Приклад застосування – українська компанія ZIR System. Компанія розробила систему розпізнавання і класифікації цілей на основі штучного інтелекту, інтегровану в безпілотні літальні апарати (БПЛА). Система може виявляти та класифікувати різні типи об'єктів, як-от транспортні засоби, будівлі та людей у режимі реального часу, що підвищує ефективність розвідувальних операцій та зменшує ризик помилок [7].

Розробка системи розпізнавання та класифікації цілей на основі штучного інтелекту. Штучний інтелект (ШІ) стає основою багатьох інновацій в області розпізнавання і класифікації об'єктів. Ці системи, засновані на використанні алгоритмів машинного навчання, гарантують автономність і точність аналізу даних і успішні в областях як-от оборони, медицини, логістики, транспорту і безпеки.

Основи розпізнавання цілей: від класичного підходу до нейронних мереж. Традиційний метод розпізнавання об'єктів базувався на обробці сигналів та зображень за допомогою алгоритму, що вимагає ручного налаштування параметрів. Наприклад, контурний аналіз або метод виявлення об'єктів на основі колірних характеристик. Однак ці підходи мали значні обмеження в складних і мінливих умовах.

Сучасні системи засновані на нейронних мережах, які дають змогу автоматизувати процес розпізнавання навчання. Наприклад [8]:

- Згорткова нейронна мережа (CNN): використовується для аналізу візуальних даних, як-от зображення та відео.
- Періодична нейронна мережа (RNN): використовується для аналізу часових рядів, як-от відеопотоки.
- Архітектура, орієнтована на реальний час: YOLO (You Only Look Once), SSD (Single Shot Multibox Detector) тощо.

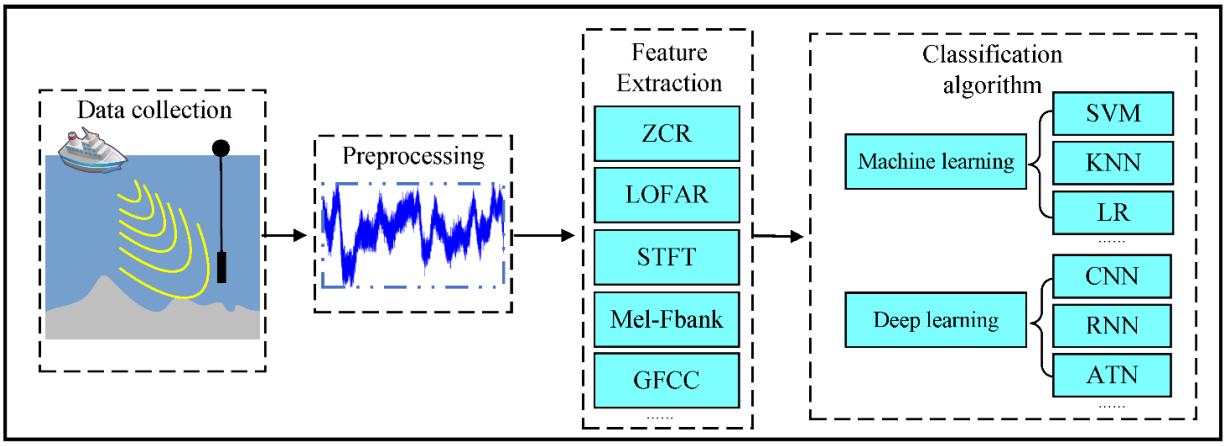


Рисунок 1 – Етапи створення системи розпізнавання та класифікації цілей

1. Збір і підготовка даних. Розробка систем на основі штучного інтелекту починається зі створення високоякісних наборів даних. Завдання розпізнавання

об’єктів вимагає 2 компонентів: великої кількості зображень або відео з різними типами цілей і анотаціями, що показує клас об’єкта (наприклад, «автомобіль», «пішохід», «дрон») і його місце розташування.

2. Модельне навчання. Коли дані готові, вибирається архітектура штучного інтелекту, придатна для навчання. Алгоритми оптимізації, як-от Adam та стохастичний градієнтний спуск (SGD), забезпечують ефективний процес навчання нейронної мережі.

3. Підтвердження. Метрики використовуються для оцінки продуктивності моделі, особливо її точності, щоб вказати, які об’єкти правильно класифіковані серед усіх ідентифікованих об’єктів і який відсоток від загальної кількості об’єктів має повноту.

4. Оптимізація. У реальних ситуаціях моделі необхідно оптимізувати для роботи з пристроями з обмеженими ресурсами, як-от вбудовані системи дронів і камери відеоспостереження.

Безумовно, можна розглянути використання не тільки у військовій сфері, а й у цивільній сфері. Насамперед системи розпізнавання на основі ШІ здатні виявляти та класифікувати ворожу техніку, позиції та навіть окремі цілі з високою точністю, наприклад [6]:

- автономні безпілотники можуть ідентифікувати пересування військової техніки;
- інтелектуальні радари інтегрують системи ШІ для швидкого виявлення повітряних та наземних загроз.

У цивільній сфері ШІ використовується здебільшого для підвищення безпеки, наприклад:

- у розумних містах для моніторингу руху транспорту та пішоходів;
- у медицині для автоматичного виявлення патологій на медичних зображеннях.

Висновки. Розробка систем розпізнавання та класифікації цілей на основі штучного інтелекту є важливим кроком у розвитку технологій автоматизації. Такі системи відкривають нові горизонти для підвищення ефективності та точності в різних сферах, включно з безпекою, логістикою, охороною здоров’я. Досягнення обчислювальної техніки та алгоритмів машинного навчання можуть зробити штучний інтелект важливим інструментом для розпізнавання та аналізу інформації в режимі реального часу.

Список використаних джерел

1. Козьяков С. Регулювання штучного інтелекту у військовій сфері: обмеження чи заохочення? *Дзеркало тижня.* 2024. URL: https://bitl.to/3G9B

2. Штучний інтелект та права людини: орієнтири та обмеження у контексті національної безпеки та оборони. *Українська гельсінська спілка з прав людини.* 2024. URL: https://bitl.to/3G9D

3. Методи та системи штучного інтелекту: Методичні вказівки та завдання для лабораторних робіт. Чернівці: ЧНУ. 2022. 38 с.

4. Шевченко Л. Штучний інтелект на війні та в армії: як використовує Китай, США та Україна. *24 канал.* 2024. URL: https://bitl.to/3G9E

5. Українська гельсінська спілка з прав людини. Використання штучного інтелекту під час війни: чи нові технології можуть порушувати права людини. 2023. URL: https://bitl.to/3G9I

6. Валкевич О. Як штучний інтелект допомагає в сучасних війнах: чи зможе він замінити військових на фронті. 2023. URL: https://bitl.to/3G9F

7. ЗІР, система розпізнавання та ураження цілей. URL: https://zir-system.com/

8. Gandhi R. R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms. Medium. 2018. URL: https://bitl.to/3G9G

УДК 004.021

Бурківський О. С., здобувач вищої освіти,
Ніколюк П. К., д-р фіз.-мат. наук,
професор, професор кафедри
інформаційних технологій

ОПТИМІЗАЦІЯ МАРШРУТІВ ПОСТАЧАННЯ ВІЙСЬКОВОЇ ТЕХНІКИ ЗА ДОПОМОГОЮ A*-АЛГОРИТМУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритм A* (A-star) – це метод пошуку шляхів у графах, який знаходить найкоротший маршрут між початковою і цільовою точками, враховуючи як пройдений шлях, так і передбачувану відстань до цілі. Завдяки поєднанню жадібного підходу та методу Дейкстри він дає змогу ефективно будувати оптимальні маршрути з урахуванням різних факторів, як-от довжина шляху, безпека чи витрати. A* широко застосовується в автомобільній навігації та ігрових системах, оскільки враховує перешкоди та забезпечує гнучкість у пошуку рішень для задач маршрутизації.

Порядок відвідування вершин у алгоритмі A* визначається за допомогою евристичної функції, яка розраховується як сума двох компонентів: вартості шляху від початкової точки до поточної вершини (позначається як g(x) і може бути як точною, так і евристичною) та оцінки відстані від поточної вершини до цільової (позначається як h(x) і є евристичною). Сума цих двох значень, позначена як f(x), дає змогу алгоритму ефективно обирати вершини для подальшого аналізу, оптимізуючи як вже пройдений маршрут, так і очікувану відстань до цільової точки.

Функція h(x) повинна бути коректною евристичною оцінкою. Наприклад, у задачах маршрутизації h(x) може відповідати відстані по прямій до цільової точки, оскільки це найменша можлива фізична відстань між двома точками [1].

Алгоритм A* послідовно досліджує всі можливі маршрути від початкової вершини до цільової, шукаючи шлях із мінімальним значенням $f(x) = g(x) + h(x)$. На кожному кроці він вибирає маршрути, які мають найвищу ймовірність досягнення мети, з урахуванням пройденого шляху g(x) і евристичної оцінки до цільової вершини h(x). Для стартової вершини алгоритм відкриває сусідні вузли з мінімальним значенням f(x) і продовжує пошук, поки значення f(x) для цільової вершини не стане найменшим або поки не будуть переглянуті всі можливі шляхи. Чим менше значення h(x), тим вищий пріоритет шляху, що дає змогу ефективно керувати чергою пріоритетів, наприклад, через використання деревоподібної структури [2].

Граф для A*-алгоритму моделюється для пошуку найкоротшого шляху між двома вузлами. Для оптимізації маршрутів постачання військової техніки за допомогою A*-алгоритму можна використовувати такий підхід:

1. Визначення вузлів графу. Вузли графу позначають ключові точки логістичної мережі. У контексті постачання військової техніки це можуть бути склади, бази забезпечення, польові штаби, прикордонні пункти або інші стратегічно важливі об’єкти.

2. Визначення ребер графу. Ребра графу моделюють транспортні шляхи між вузлами. Це можуть бути дороги, залізничні маршрути, річкові чи повітряні коридори, які використовуються для перевезення техніки або вантажів.

3. Призначення ваги ребрам. Вага ребер відображає витрати на переміщення між вузлами. Можна враховувати фактори, як-от відстань, час у дорозі, ризикованість маршруту, витрати пального, стан доріг або загрози (наприклад, можливість ворожих атак) [3].

Після цього потрібно визначити початковий і кінцевий вузол, щоб знайти найкоротший шлях між ними. Можемо розглянути використання A*-алгоритму лінії фронту, зокрема для знаходження найкоротшого шляху і найбезпечнішого маршруту з позиції J до вершини D. Візьмемо за початкову вершину J як стартову, і у якості кінцевої вершини D (рис. 1).

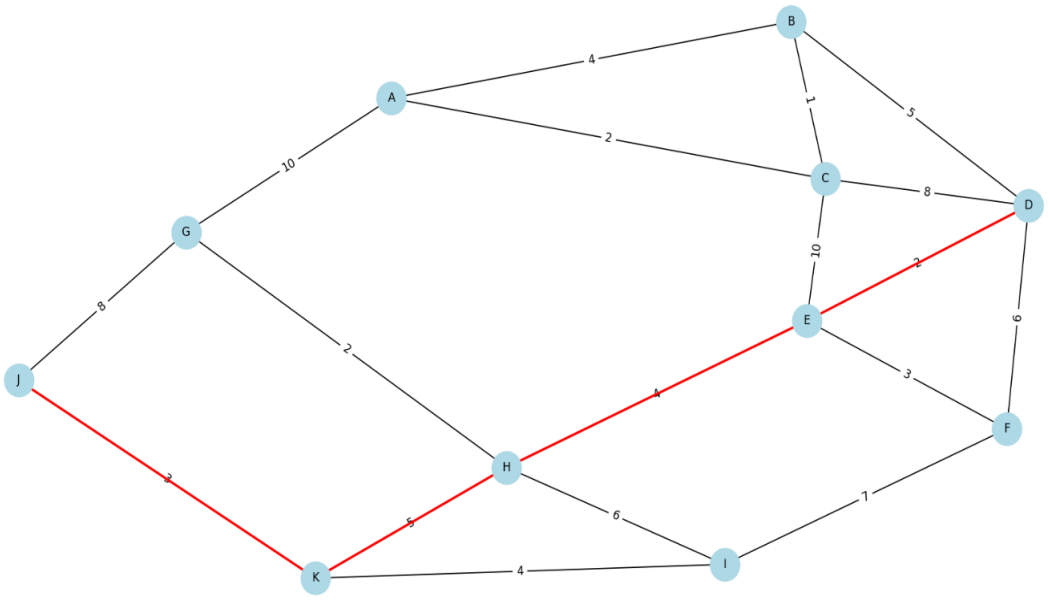


Рисунок 1 – A-алгоритм міського трафіка м. Вінниці*

Потім розраховуємо в програмі Thony алгоритм постачання військової техніки на лінію фронту, а саме найкоротший шлях від початкової вершини J до кінцевої вершини D (рис. 2). Під час роботи програми враховуються всі визначені параметри, як вага ребер графу, евристична оцінка h(x), а також можливі перешкоди на маршруті. Наприклад, у реальних умовах війни цей алгоритм дає змогу врахувати заблоковані ділянки доріг, загрозу обстрілів або інші небезпеки. Завдяки гнучкості A* можливо динамічно переналаштовувати маршрут у відповідь на зміну умов, наприклад, у разі перекриття шляху або появи більш безпечного маршруту.

Програма може також інтегрувати додаткові модулі для аналізу ризиків, враховуючи інформацію з дронів, супутників або інших джерел розвідки. Наприклад, врахування ризику обстрілу на певних ділянках може бути реалізовано шляхом збільшення ваги відповідних ребер графу. Такі можливості роблять алгоритм A* надзвичайно ефективним у військових та інших критичних застосуваннях.

```python
def a_star(graph, start, goal, heuristic):
    open_set = PriorityQueue()
    open_set.put((0, start))
    came_from = {}
    g_score = {node: float('inf') for node in graph.nodes}
    g_score[start] = 0
    f_score = {node: float('inf') for node in graph.nodes}
    f_score[start] = heuristic[start]

    while not open_set.empty():
        current = open_set.get()[1]

        if current == goal:
            path = []
            while current in came_from:
                path.append(current)
                current = came_from[current]
            path.append(start)
            path.reverse()
            return path

        for neighbor in graph.neighbors(current):
            tentative_g_score = g_score[current] + graph[current][neighbor]['weight']
            if tentative_g_score < g_score[neighbor]:
                came_from[neighbor] = current
                g_score[neighbor] = tentative_g_score
                f_score[neighbor] = g_score[neighbor] + heuristic[neighbor]
                open_set.put((f_score[neighbor], neighbor))

    return None
```

Рисунок 2 – Код A-алгоритму*

Результат роботи програми виглядає так [J, K, H, E, D].

Підсумовуючи, можна сказати, що A*-алгоритм є корисним інструментом для оптимізації маршрутів постачання військової техніки. Цей алгоритм дає змогу визначити найефективніший шлях між стратегічно важливими точками, враховуючи різні фактори, що впливають на логістику, як-от відстань, час у дорозі, витрати ресурсів та рівень ризику. Використання A*-алгоритму допомагає мінімізувати час доставки, зменшити витрати на транспортування і підвищити безпеку та ефективність постачання в умовах військових операцій.

Список використаних джерел

1. Hart P., Nilsson N., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics. Vol. 4, iss. 2. July 1968. P. 100–107. URL: https://ieeexplore.ieee.org/document/4082128 (дата звернення: 23.11.2024).

2. Zeng W. Church R. Finding Shortest Paths on Real Road Networks: The Case for A*. *International Journal of Geographical Information Science.* 2009. URL: https://www.tandfonline. com/doi/abs/10.1080/13658810903270559 (дата звернення: 23.11.2024).

3. Dechter R., Pearl J. Generalized Best-First Search Strategies and the Optimality of A*. Journal of the ACM. URL: https://dl.acm.org/doi/10.1145/214418.214421 (дата звернення: 23.11.2024).

УДК 004.67

Дорофєєв Є. О., здобувач вищої освіти,
Січко Т. В., канд. техн. наук, доцент,
доцент кафедри інформаційних технологій

РОЛЬ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ (BIG DATA) У СУЧАСНИХ ІНОРМАЦІЙНИХ СИСТЕМАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі дані стають новою «валютою». Розвиток цифрових технологій, Інтернету речей (IoT) і глобальної взаємодії з інформаційними системами призводить до стрімкого зростання обсягів інформації. Аналіз великих даних (Big Data) дає змогу трансформувати структуровані масиви даних у цінну інформацію, яка використовується для ухвалення рішень у бізнесі, науці, державному управлінні та інших сферах. Впровадження Big Data в інформаційні системи стало ключовим фактором їх ефективності.

Актуальність теми зумовлена необхідністю обробки та аналізу великих обсягів даних [1]. Великі дані стали основою для впровадження інновацій у різних галузях – від персоналізованої медицини до автоматизації бізнес-процесів. Проте ефективне використання Big Data вимагає вирішення низки викликів, зокрема забезпечення продуктивності систем, захисту конфіденційності даних та інтеграції наявними інформаційними системами.

Постановка завдання:

- проаналізувати концептуальні основи Big Data та їх роль у сучасних інформаційних системах;
- дослідити підходи до інтеграції аналітики великих даних у бізнес-процеси;
- розглянути переваги та виклики використання Big Data.

Аналітика великих даних (Big Data Analytics) є ключовим компонентом сучасних інформаційних систем, який допомагає працювати з великими обсягами різнорідної інформації. Основні технології, які використовуються в аналітиці, включають платформи для обробки даних у розподілених середовищах (Hadoop, Apache Spark), інструменти візуалізації (Tableau, Power BI) і алгоритми машинного навчання, що дають змогу аналізувати як структуровані, так і неструктуровані дані [2].

Одним із найважливіших напрямів використання аналітики великих даних є оптимізація бізнес-процесів. Завдяки аналізу даних з різних джерел (логістичних систем, клієнтських баз, соціальних мереж) компанії можуть виявляти неефективності у своїй роботі та швидко реагувати на зміни попиту.

Наприклад, Walmart використовує Big Data для аналізу сезонного попиту, що допомагає значно скоротити витрати на логістику та зберігання товарів [3].

Іншим важливим аспектом є персоналізація послуг, яка базується на аналізі поведінкових даних користувачів. Великі платформи, як-от Netflix, Amazon або Spotify, створюють індивідуальні рекомендації для своїх користувачів, використовуючи складні алгоритми аналізу великих даних. Це дає змогу не лише підвищувати лояльність клієнтів, але й збільшувати дохід компаній завдяки точному таргетуванню.

У виробництві аналітика великих даних сприяє прогнозуванню поломок обладнання (Predictive Maintenance), використовуючи дані про роботу машин і технічні параметри. Такий підхід допомагає запобігати непередбаченим аваріям, знижувати витрати на ремонт і підвищувати ефективність виробничих процесів. Наприклад, компанії в автомобільній промисловості використовують ці дані для моніторингу роботи двигунів та інших важливих компонентів.

У сфері охорони здоров'я Big Data використовуються для виявлення прихованих закономірностей у даних про пацієнтів, як-от залежності між генетичними факторами та ефективністю ліків. Це відкриває нові можливості для персоналізованої медицини, а також дає змогу ефективніше прогнозувати спалахи епідемій і керувати медичними ресурсами.

Також важливо відзначити роль Big Data у реалізації концепції «розумних міст». Аналіз даних, отриманих від сенсорів IoT, дає змогу оптимізувати трафік, зменшувати енергоспоживання, поліпшувати екологічний стан міст і підвищувати рівень життя мешканців. Наприклад, міста Сінгапур і Барселона активно використовують аналітику для керування транспортною інфраструктурою та енергомережами.

У державному секторі великі дані допомагають боротися з шахрайством, виявляючи аномалії у фінансових операціях або податкових звітах, та підвищують прозорість управління завдяки аналізу відкритих даних. До того ж аналітика використовується для прогнозування соціально-економічних явищ, як-от рівень безробіття, ріст інфляції або вплив політичних рішень на економіку.

Big Data забезпечує:

- підвищення точності прогнозів;
- автоматизацію процесів і зменшення часу на обробку інформації;
- поліпшення взаємодії з клієнтами завдяки персоналізації.

Проте виклики з конфіденційністю, масштабуванням та браком кваліфікованих кадрів залишаються актуальними.

Отже, аналітика великих даних стає основою для прийняття стратегічних рішень у багатьох галузях. Її використання дає змогу не лише підвищувати ефективність, але й створювати нові бізнес-моделі, які відповідають вимогам цифрової епохи. Однак успішна інтеграція цих технологій потребує подолання викликів, як-от забезпечення безпеки даних, вирішення проблем масштабованості та залучення висококваліфікованих фахівців.

Список використаних джерел

1. Hadoop Apache Project. Hadoop Framework Documentation. Опис технології обробки великих даних у розподілених системах. URL: https://hadoop.apache.org/docs/stable/ (дата звернення: 28.11.2024).

2. Dean J., Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM. 2008. URL: https://static.googleusercontent.com/media/research.google.com/uk//archive/mapreduce-osdi04.pdf (дата звернення: 28.11.2024).

3. Walmart Global Tech. Using Data to Enhance Supply Chain Efficiency. URL: https://tech.walmart.com/content/walmart-global-tech/en_us/blog/post/walmarts-ai-powered-inventory-system-brightens-the-holidays.html (дата звернення: 28.11.2024).

4. Ткачук Н. О., Січко Т. В. Застосування Від Data у бізнесі. *Комп'ютерні технології обробки даних:* матеріали всеукр. наук.-практ. конф. Вінниця, 2022. С. 224–226.

5. Підруцький Д. А., Січко Т. В. Big Data в обробці і аналізі даних з різних сфер. *Комп'ютерні технології обробки даних:* матеріали всеукр. наук.-практ. конф. Вінниця, 2023. С. 183–187.

УДК 681.327.8

Поліщук В. С., здобувач вищої освіти.
Ніколюк П. К., д-р фіз.-мат. наук,
професор, професор кафедри
інформаційних технологій

РОЗРОБКА МОДЕЛІ АКУСТИЧНОГО ЛОКАТОРА ДЛЯ ВИЯВЛЕННЯ ДРОНІВ СУПРОТИВНИКА НА НИЗЬКИХ ВИСОТАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

На тлі сучасної воєнної агресії проти України однією з ключових проблем національної системи протиповітряної оборони є ефективне виявлення низьковисотних повітряних цілей супротивника, як-от безпілотні літальні апарати (БПЛА) і крилаті ракети. Ці загрози характеризуються малою радіолокаційною помітністю, маневреністю, використанням важкопрохідної місцевості для запобігання виявлення і зниження ефективності звичайних радіолокаційних систем. У таких ситуаціях багатообіцяльним рішенням стають акустичні локатори, засновані на розподілених мікрофонних мережах. Завдяки здатності виявляти звукові хвилі від двигуна і аеродинамічні шуми така система дає змогу точно визначати місце розташування повітряних цілей навіть у складних ситуаціях [1]. Розробка моделей акустичних локаторів для виявлення повітряних цілей передбачає створення ефективної мережі мікрофонів, оптимально розміщених на вишках зв'язку для забезпечення максимального охоплення місцевості. Для точного визначення координат джерела звуку використовується алгоритм розрахунку часу, за який звукові хвилі досягають кожного мікрофона. Це допомагає обробляти дані в режимі реального часу з урахуванням впливу шуму, погодних умов та інших перешкод. Розроблена модель повинна бути інтегрована з наявною системою протиповітряної оборони, що дасть змогу своєчасно передавати координати виявлених цілей для їх оперативного знищення [2].

Використання акустичних локаторів особливо ефективно під час виявлення цілей з низькою радіолокаційною помітністю, як-от безпілотні літальні апарати, оскільки система не залежить від рельєфу місцевості або наявності перешкод, що обмежують роботу радара. До того ж використання наявної телекомуні-

каційної інфраструктури знижує витрати на впровадження системи. Створення такої моделі забезпечить підвищення ефективності виявлення авіаційних цілей, скорочення часу реагування систем протиповітряної оборони та підвищення рівня безпеки повітряного простору України. Впровадження акустичних локаторів могло б стати важливим кроком у зміцненні обороноздатності держави захисту населення і критично важливої інфраструктури від сучасних загроз з повітря [3].

Очікуючи результатів під час розробки моделі акустичного локатора, ми можемо ефективно інтегрувати акустичний локатор з наявними системами протиповітряної оборони і забезпечити своєчасне реагування на загрози. Розробка математичних моделей і алгоритмів, що допомагають виявляти повітряні цілі з похибками в межах допустимих значень для військових систем, а також підвищення здатності ППО України ефективно діяти проти супротивника в умовах складної обстановки, зокрема у випадках інтенсивного використання БПЛА відображені на рис. 1.

```python
import numpy as np
from scipy.optimize import minimize
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

# --------------------------
# 1. Вхідні дані
# --------------------------

# Координати телекомунікаційних вишок (мікрофонів)
mic_positions = np.array([
    [0, 0],       # Вишка 1
    [0, 100],     # Вишка 2
    [100, 0],     # Вишка 3
    [100, 100]    # Вишка 4
])

# Справжні положення кількох джерел звуку (дрони)
targets = [
    np.array([50, 75]),  # Дрон 1
    np.array([30, 20]),  # Дрон 2
    np.array([80, 90])   # Дрон 3
]

# Швидкість звуку (м/с)
speed_of_sound = 343  # при 20°C

# Радіус дії мікрофонів (у метрах)
mic_range = 120

# --------------------------
# 2. Розрахунок часу прибуття сигналу
```

Рисунок 1 – Частина коду акустичного локатора

Цей код демонструє можливість акустичного локатора для виявлення низьковисотних повітряних цілей, як-от безпілотні літальні апарати. Реалізація додає імітацію реальної роботи мікрофона з урахуванням його дальності дії, шуму сигналу, динамічного переміщення цілі, а також перевірки небезпеки і передачі інформації в систему ППО.

1. Візуалізуємо роботу системи акустичного локатора. Ви можете побачити, як мікрофон фіксує рух дронів і як оцінює їх місцезнаходження.

2. Змоделюємо рух дронів. Цілі рухаються по заданих траєкторіях, і система постійно оновлює їх координати.

3. Виявляємо небезпечні об’єкти. Коли безпілотник наближається до центру (наприклад, критичної зони), система попереджає про небезпеку.

4. Передаємо дані на ППО. Інформація про виявлені цілі передається умовно на засоби протиповітряної оборони.

Нижче наведено зображення роботи цього коду, на якому демонструється, як система виявляє безпілотники, оцінює їх місцезнаходження та відображає їх у реальному часі (рис. 2–3).

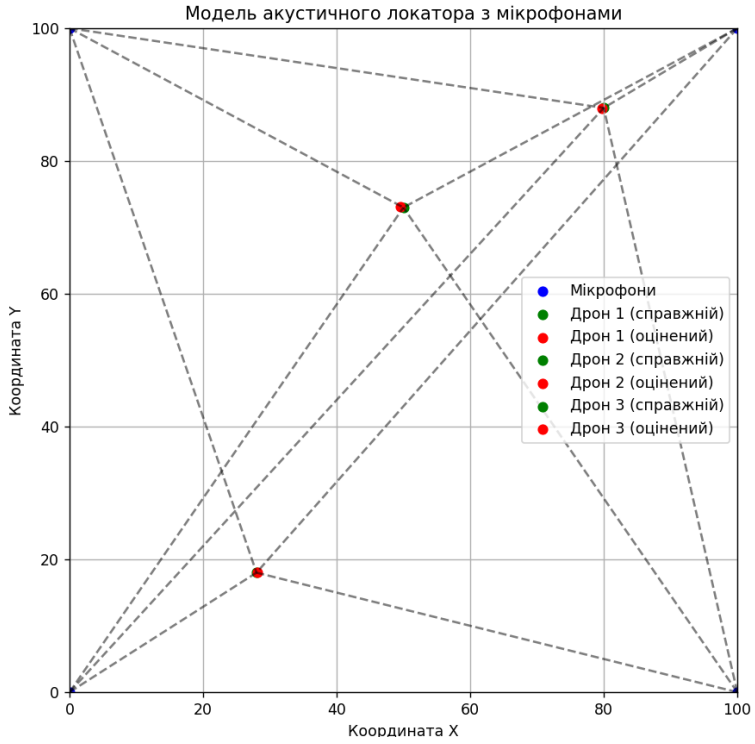


Рисунок 2 – Акустичний локатор для виявлення дронів

```
Попередження! Ворожий дрон на відстані 25.20 м.
Передача даних на ППО:
- Координати дрона: [49.79623496 75.19756735]
- Координати дрона: [29.78084751 19.72523615]
- Координати дрона: [79.82194196 89.52580434]
Інформація успішно передана!
```

Рисунок 3 – Результати акустичного локатора для виявлення дронів

Розробка та впровадження акустичних локаторів зробить значний внесок у захист неба в Україні, допоможуть знизити жертви серед цивільного населення та критично важливої інфраструктури, а також зміцнять обороноздатність країни. Створюючи мобільні та стаціонарні акустичні засоби на основі розробленої моделі, її можна адаптувати до різних умов, особливо в місцях, де немає стаціонарної інфраструктури. Впровадження цієї системи стане важли-

вим кроком для зміцнення обороноздатності України та забезпечення безпеки національного повітряного простору.

Список використаних джерел

1. Технології виявлення та протидії дронам на сьогоднішній день. *Bezpeka.club*. 2022. URL: https://bezpeka.club/10-technologies-for-detecting-countering-drones (дата звернення: 24.11.2024).

2. Development of an Acoustic System for UAV Detection. MDPI. URL: https://www.mdpi.com/1424-8220/20/17/4870 (дата звернення: 24.11.2024).

3. Low-Altitude UAV Surveillance System via Highly Sensitive Distributed Acoustic Sensing. IEEEXplore. URL: https://ieeexplore.ieee.org/document/10667006 (дата звернення: 24.11.2024).

УДК 004.8:711.4:656

Семен О. Д., здобувач вищої освіти,
Ніколюк П. К., д-р фіз.-мат. наук,
професор, професор кафедри
інформаційних технологій

ІНТЕГРАЦІЯ ШТУЧНОГО ІНТЕЛЕКТУ У ТРАНСПОРТНІ СИСТЕМИ ДЛЯ ОПТИМІЗАЦІЇ МІСЬКОЇ ІНФРАСТРУКТУРИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Зі зростанням урбанізації міста стикаються з проблемами перевантаження транспортних мереж, забруднення довкілля та неефективного використання інфраструктури. Згідно з дослідженнями, у великих містах водії витрачають до 30 % часу у заторах. Інтеграція штучного інтелекту у транспортні системи дає змогу автоматизувати управління потоками, прогнозувати завантаженість доріг, оптимізувати маршрути та зменшити екологічний вплив.

Рисунок 1 – Приклад роботи системи передбачення заторів

Наукові роботи останніх років демонструють значний прогрес у застосуванні штучного інтелекту для вирішення транспортних проблем. Дослідники з Массачусетського технологічного інституту розробили систему на основі глибинного навчання, яка може прогнозувати затори за 30 хвилин до їх виникнення з точністю 85 % [1].

У Європі активно впроваджуються розумні світлофори, що адаптуються до реальної інтенсивності руху, зменшуючи час очікування на перехрестях до 20 %. В Україні поки що перспективи застосування таких технологій залишаються недооціненими, що підкреслює потребу у нових дослідженнях та впровадженнях.

Оскільки у впровадженні ініціатив велику роль відіграє виділений бюджет, то рішення проблеми має бути максимально ефективним, за мінімальну вартість. У реаліях України можна було б використовувати розумні світлофори.

Рисунок 2 – Візуалізація системи інтелектуальних світлофорів

Для роботи такого світлофора потрібне створення точних прогнозів, тому алгоритм має бути досконалим і використовувати різні джерела даних:

1. Датчики руху: на ключових вузлах можна встановити камери, які будуть навчені збирати інформацію про кількість автомобілів, їх швидкість, визначати завантаженість дороги. Розвиток штучного інтелекту дає змогу втілити це в реальність.

2. GPS-дані: інформація з навігаторів автомобілів і мобільних додатків забезпечила б відстеження місцезнаходження транспорту, середньої швидкості руху, часу затримок у реальному часі.

3. Статистичні дані: вищезгадана модель аналізувала історію аварій, карти доріг, супутникові зображення та траєкторії GPS. Система міського трафіка динамічна, але має певні закономірності, хоч і з певними відхиленнями.

Використовуючи ці дані, алгоритм може виявляти патерни у динаміці транспортних, зокрема:

1) часові тренди: аналізуючи циклічні зміни у завантаженості доріг, наприклад, ранкові та вечірні пікові години;

2) просторові закономірності: визначаючи критичні точки транспортної мережі, як-от перехрестя з регулярними заторами чи ділянки з високим навантаженням та аварійністю;

3) реакція на події: оцінюючи вплив раптових змін, наприклад погіршення погоди або аварії.

На цьому етапі можна використовувати згорткові нейронні мережі (Convolutional Neural Networks, CNN), тому що вони працюють із просторовими даними, включно з супутниковими знімками та картами доріг [2].

Наступним етапом після аналізу даних є прогнозування. На етапі прогнозування використовуються рекурентні нейронні мережі (Long Short-Term Memory, LSTM), оскільки вони мають такі переваги:

1. LTSM були спеціально розроблені для врахування попередніх станів у послідовних даних. У транспортній системі теперішній стан залежить від попередніх станів, наприклад, рівня трафіка чи пікових годин. LSTM зберігають контекст попередніх даних через свої «комірки пам'яті», що дає їм змогу прогнозувати майбутній стан дорожнього руху точніше.

2. LTSM фільтрують нерелевантну інформацію за допомогою механізму «forget gate», залишаючи ключові фактори.

3. LTSM здатні вивчати регулярні змінні на основі історичних даних, а також швидко адаптуватися до нових ситуацій, використовуючи поточні дані.

4. LTSM можуть інтегрувати дані з різних джерел, та враховувати їх часовий контекст. Наприклад, дані про сповільнення руху можуть бути поєднані з погодними умовами, щоб зробити точніший прогноз.

LSTM вже використовуються у транспортних дослідженнях, тому вони є ідеальним інструментом для прогнозування стану транспортної системи, оскільки вони здатні враховувати як короткострокові, так і довгострокові закономірності в даних, забезпечуючи адаптивність і точність у мінливих умовах [3].

Отримавши прогнозовані дані, наступним етапом буде розподіл пріоритетів. На цьому етапі алгоритм може виконувати такі дії, залежно від отриманих даних:

1. Критичні перехрестя: отримавши дані про трафік, алгоритм ідентифікуватиме перехрестя як критичне, та регулюватиме час роботи світлофорів зеленого та червоного кольору.

2. Пріоритетний рух: система може надавати перевагу громадському транспорту чи автомобілям екстрених служб, зменшуючи їх час затримки. Такі дії позитивно вплинуть на багато процесів, від якості роботи екстрених служб до зменшення кількості викидів, стимулюючи людей користуватися громадським транспортом.

3. Ітеративний підхід: алгоритм безперервно оновлює рішення, використовуючи дані, що надходять у реальному часі. Це забезпечить ефективну роботу всієї системи.

Загалом використання штучного інтелекту допоможе автоматизувати управління потоками транспорту. Така ініціатива позитивно вплине на величезну кількість явищ, серед яких час простою в заторах, збільшення пропускної здатності мереж, зменшення кількості викидів в атмосферу, зменшення кількості аварій, зниження стресу водіїв. Цей напрям має перспективу для досліджень, можна надалі оптимізувати алгоритми його роботи, знаходити дешевші альтернативи для втілення ідеї, створювати умови для використання цієї ідеї. Головною метою досліджень буде адаптація підходу до реальних міських умов, оскільки в симуляції він показує себе добре.

Список використаних джерел

1. Gordon R. Deep learning helps predict traffic crashes before they happen. *MIT Schwarzman College of Computing Massachusetts.* 2021. URL: https://computing.mit.edu/news/deep-learning-helps-predict-traffic-crashes-before-they-happen/ (дата звернення: 28.11.2024).
2. Згорткова нейронна мережа. *Вікіпедія.* URL: https://uk.wikipedia.org/wiki/Згорткова_нейронна_мережа (дата звернення: 28.11.2024).
3. Long Short-Term Memory (LSTM). *IT Wiki*. 2023. URL: https://itwiki.dev/data-science/ml-reference/ml-glossary/long-short-term-memory-lstm (дата звернення: 28.11.2024).

УДК 004.8

Слободянюк С. С., здобувач вищої освіти,
Якубич К. О., асистент кафедри
інформаційних технологій

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В СИСТЕМНОМУ МОДЕЛЮВАННІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Актуальність теми. У сучасному світі складність технічних, економічних, соціальних та екологічних систем постійно зростає, що вимагає нового підходу до їх моделювання та аналізу. Традиційні методи системного моделювання часто виявляються неефективними через велику кількість даних, багатофакторність та необхідність швидкого прийняття рішень.

Штучний інтелект (ШІ) стає невід'ємною частиною цього процесу, надаючи інструменти для автоматизації, адаптивності та прогнозування. Його застосування дає змогу створювати більш точні, динамічні та адаптивні моделі, здатні враховувати складні взаємозв'язки всередині системи. Враховуючи широкий спектр застосування ШІ в галузях від медичної до промислової, вивчення можливостей ШІ в системному моделюванні дуже важливе.

Використання ШІ в системному моделюванні:

1. Автоматизація моделювання. Штучний інтелект допомагає автоматизувати процес аналізу великих обсягів даних і створення моделей. Наприклад, алгоритми машинного навчання можуть генерувати моделі систем без необхідності ручного програмування, що значно прискорює процес.

2. Оптимізація складних систем. Методи оптимізації на основі штучного інтелекту, як-от генетичні алгоритми та технологія рою частинок, можуть допомогти вам знайти найкращі рішення складних проблем. Це особливо корисно, якщо система має безліч змінних і параметрів, що впливають на її роботу.

3. Прогнозування та аналіз сценаріїв. За допомогою нейронних мереж та методів глибокого навчання ви можете створювати прогнозні моделі для оцінки поведінки системи в різних умовах. Це важливо для аналізу ризиків, розробки стратегії та прийняття рішень.

4. Створення адаптивних моделей. Інтеграція AI дає змогу створювати адаптивні моделі, які змінюються в режимі реального часу у відповідь на нові

дані. Такі моделі здатні більш ефективно реагувати на зміни у зовнішньому середовищі.

5. Цифрові двійники. Одним із найбільш перспективних напрямів є розробка цифрових двійників (віртуальних копій фізичних об'єктів або систем). Цифровий двійник допомагає проводити експерименти, перевіряти гіпотези і аналізувати поведінку системи, не піддаючи ризику реальні об'єкти.

Проблеми застосування штучного інтелекту під час моделювання систем:

1. Складність обчислень. Великі обсяги даних і складні алгоритми вимагають значних ресурсів.

2. Якість даних. Для створення точної моделі потрібні чисті, структуровані та актуальні дані.

3. Інтерпретованість. Моделі штучного інтелекту часто є «чорними ящиками», і користувачам може бути важко зрозуміти, як вони працюють.

Перспективи розвитку ШІ в системному моделюванні:

1. Розвиток квантових обчислень. Квантові комп'ютери обіцяють значно розширити можливості обробки даних та виконання складних обчислень. Інтеграція квантового обчислення зі штучним інтелектом допоможе вирішувати завдання моделювання, які наразі потребують величезних ресурсів. Це відкриває перспективи для більш точного прогнозування та моделювання систем у реальному часі.

2. Удосконалення нейронних мереж. Подальший розвиток архітектур нейронних мереж, як-от трансформери та моделі глибокого навчання, дасть змогу створювати ще більш точні прогнози та адаптивні системи. Ці технології забезпечать глибший аналіз складних взаємозв'язків у системах, які були недоступні для традиційних методів.

3. Автоматизація моделювання та проєктування. Штучний інтелект стане основою для автоматизованих платформ моделювання. Такі платформи допоможуть швидко створювати моделі без необхідності втручання експертів, що спрощує використання технології для широкого кола користувачів, включно з інженерами, бізнесменами і науковцями.

4. Інтеграція з візуалізацією та віртуальною реальністю (VR). ШІ у поєднанні з VR відкриє нові можливості для візуалізації моделей. Це дасть змогу користувачам взаємодіяти з моделями в інтерактивному середовищі, експериментувати з різними сценаріями та краще розуміти складні системи.

Висновки. Штучний інтелект стає важливим інструментом системного моделювання, пропонуючи нові можливості для аналізу, прогнозування та оптимізації складних систем. Його використання допомагає підвищити ефективність і точність моделювання та відкриває перспективи для вирішення глобальних проблем у різних галузях. Але для успішного впровадження ШІ необхідно подолати проблеми, пов'язані з ресурсами, даними та інтерпретацією.

Список використаних джерел

1. Unite Ai. URL: https://www.unite.ai/uk/ (дата звернення: 01.12.2024).

2. IT Enterprise. URL: https://www.it.ua/knowledge-base/technology-innovation/cifrovoj-dvojnik-digital-twin (дата звернення: 01.12.2024).

3. Ansys. URL: https://www.ansys.soften.com.ua/about-ansys/blog/749-poednannya-shtuchnogo-intelektu-ta-modelyuvannya-u-ansys.html (дата звернення: 01.12.2024).

УДК 004.93

Щербина Д. С., здобувач вищої освіти,
Ніколюк П. К., д-р фіз.-мат. наук,
професор, професор кафедри
інформаційних технологій

ЗАСТОСУВАННЯ МОДЕЛІ YOLO ДЛЯ РОЗПІЗНАВАННЯ ВОРОЖОЇ ВІЙСЬКОВОЇ ТЕХНІКИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні умови ведення бойових дій передбачають постійне використання передових технологій у військовій сфері, а автоматизоване розпізнавання ворожої техніки є важливим елементом захисту і реагування. Враховуючи динамічний характер сценаріїв бойових дій та обмежений час для прийняття рішень, використання алгоритмів комп'ютерного зору стає невід'ємною частиною сучасних систем захисту. Модель You Only Look Once (YOLO) відома своєю високою швидкістю та точністю в задачах розпізнавання об'єктів і демонструє великий потенціал для вирішення цієї проблеми.

YOLO (You Only Look Once) – це потужна модель для розпізнавання об'єктів у зображеннях і відео, яка вирізняється високою швидкістю та точністю. Вона відрізняється від традиційних підходів тим, що не розділяє процес визначення областей об'єктів і їх класифікації, а виконує ці завдання одночасно, аналізуючи все зображення за один прохід через нейронну мережу.

Модель працює як єдина структура, сприймаючи зображення як сітку, у якій кожна клітинка відповідає за виявлення об'єктів і визначення їх класів. Завдяки своїй архітектурі YOLO здатна обробляти кадри з високою швидкістю, що робить її ідеальною для застосування в реальному часі. До того ж аналіз усього зображення дає змогу моделі враховувати контекст, що значно зменшує кількість хибних розпізнавань. Також завдяки своїй гнучкості YOLO може легко адаптуватися до нових задач, якщо вона проходить додаткове навчання на відповідних наборах даних. Саме ці риси роблять її популярною в багатьох сферах, особливо у військовій [1].

Щоб розробити й оптимізувати систему, яка буде автоматично розпізнавати ворожу військову техніку на основі моделі YOLO, а також буде враховувати додаткові потреби розпізнавання, потрібно виконати такі умови:

- підготувати датасет, який буде включати великий набір зображень різних типів техніки, як-от танки, БТР, БМП, артилерійські установки та вантажівки. Також варто підібрати зображення з різними погодними умовами, умовами освітлення, кутами огляду та якістю. Окрім зображень, потрібно створити класи для всіх типів об'єктів датасету, а також зазначити складні випадки, як-от накриття камуфляжем чи макет техніки [2];
- під час налаштування і навчання моделі доцільно використовувати версії YOLOv5 або YOLOv8. Версія YOLOv8 більш сучасна, тому в неї краща точність, сучасніший функціонал, а також вона може працювати з більш складними наборами даних, проте це є ресурсомістким процесом. Тому якщо потрібна

модель, яка буде ефективна для більшості задач, хоча і з меншою точністю, тоді можна використати версію YOLOv5. Окрім вибору версії, потрібно правильно налаштувати параметри, а саме: адаптувати для роботи з низькоякісними зображеннями, імітувати реальні умови, як-от розмиття та часткове перекриття об’єктів [3].

Після виконання даних умов потрібно провести тестування моделі на даних, які не брали участь у навчанні моделі. Достатніми умовами для подальшого вдосконалення моделі є більше 80 % точності для гарно видимих об’єктів та більше 60 % для об’єктів з різними погіршеннями якості зображення.

За достатніх умов точності можна розглядати подальше вдосконалення моделі, як-от додавання інфрачервоних та радіолокаційних зображень. До того ж можна вдосконалити систему для роботи з відеопотоками в реальному часі.

Підсумовуючи, можна сказати, що застосування моделі YOLO у військовій сфері може стати ефективним рішенням для автоматизованого розпізнавання ворожої техніки. Також її впровадження у військові технології може покращити ситуаційну обізнаність та сприяти оперативнішому прийняттю рішень.

Список використаних джерел

1. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, 2016. С. 779–788.

2. Онищенко С., Лактіонов О., Глушко А. Використання штучного інтелекту для розпізнавання терористичних та ворожих військових об’єктів. *Herald of Khmelnytskyi National University*. Technical sciences. 2024. Vol. 335.3(1). С. 166–171.

3. Hussain M. Yolov5, yolov8 and yolov10: The go-to detectors for real-time vision. arXiv preprint arXiv:2407.02988. 2024.

СЕКЦІЯ 3
ЕКСПЕРТНІ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ ТА СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

УДК 004.89:371.26

Чуприна І. А., здобувач вищої освіти,
Антонов Ю. С., канд. фіз.-мат. наук, доцент,
доцент кафедри інформаційних технологій

ОСОБЛИВОСТІ РОЗРОБКИ СИСТЕМИ АВТОМАТИЧНОГО СТВОРЕННЯ ТА ВАЛІДАЦІЇ ТЕСТОВИХ ПИТАНЬ НА ОСНОВІ ВЗАЄМОДІЇ З CHATGPT

Донецький національний університет імені Василя Стуса, м. Вінниця

Актуальність дослідження. У сучасному світі інформаційних технологій та диджиталізації освітніх процесів автоматизація створення та оцінювання навчальних матеріалів стає важливим інструментом для підвищення ефективності навчання [1–3]. Традиційні методи розробки тестових завдань часто є трудомісткими і вимагають значних витрат часу з боку викладачів. З іншого боку, штучний інтелект (ШІ) дає змогу автоматизувати ці процеси, забезпечуючи швидку розробку якісних тестових завдань. Це не тільки скорочує час на підготовку тестів, але й допомагає адаптувати навчальні матеріали під конкретні потреби студентів.

Система автоматичного створення та валідації тестових питань на основі ChatGPT (https://openai.com/chatgpt) відкриває нові можливості для навчальних закладів. Вона здатна забезпечити більш точне оцінювання знань, адаптуючи завдання під рівень студентів і мінімізуючи людський фактор у процесі розробки тестів. Така система може бути інтегрована у сучасні освітні платформи, що дасть змогу розширити її використання і в дистанційній освіті. Особливе значення цієї технології полягає в тому, що вона може суттєво змінити підходи до оцінювання, підвищуючи їх об’єктивність і адаптивність. Актуальність теми підтверджується зростаючим попитом на автоматизовані інструменти в освіті, особливо в умовах глобалізації та переходу до дистанційного навчання.

Мета та завдання дослідження. Метою цієї роботи є створення програмного забезпечення, яке автоматично генерує та оцінює тестові питання на основі взаємодії з моделлю ChatGPT. Система повинна забезпечувати високу точність і релевантність питань, що відповідають навчальній програмі, а також проводити оцінку складності завдань відповідно до когнітивних і аналітичних здібностей студентів. Це допоможе забезпечити адаптивність тестів до різних рівнів знань та індивідуальних потреб кожного студента.

До основних завдань дослідження входить аналіз наявних алгоритмів генерації текстових завдань, їх адаптація до вимог сучасної освіти, а також дослідження можливостей налаштування параметрів штучного інтелекту для досягнення максимальної точності. До того ж необхідно оцінити ефективність використання моделі ChatGPT для автоматизації тестування, що включає порівняння отриманих результатів із традиційними методами. Під час дослідження також важливо визначити можливості інтеграції системи в різні освітні платформи, зокрема ті, що використовують мобільні додатки або вебверсії для тестування студентів.

Методологія. У роботі використовувалася мова програмування C#, яка є потужним інструментом для розробки вебсервісів на базі технології Web API з прив'язкою до штучного інтелекту. Основною бібліотекою, яка була використана, є Betalgo.Ranul.OpenAI(https://github.com/betalgo/openai) – інструмент, що забезпечує легку інтеграцію з мовними моделями GPT-3 та ChatGPT. Це дало змогу організувати процес взаємодії між програмою і моделлю ШІ, забезпечуючи автоматизацію генерації питань на основі введеного тексту.

Оскільки система призначена для роботи в університеті, в якому існує багато сутностей, пов'язаних між собою, то першим кроком була продумана база даних, яка має схему, продемонстровану на рис. 1.

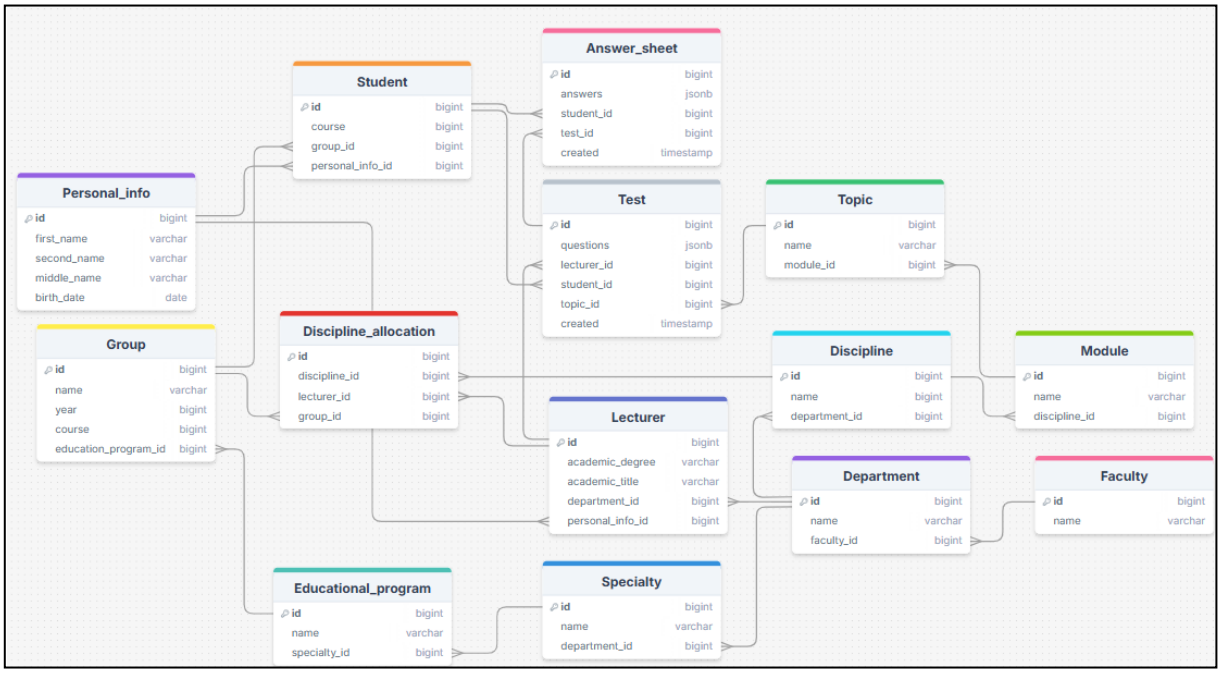


Рисунок 1 – Схема бази даних

Наступним кроком була продумана основна логіка, а саме взаємодія з CharGPT для того, щоб генерувати тестові питання з множинним вибором. Оскільки напряму неможливо комунікувати з цим ШІ через програмний код, для цього був використаний OpenAI API. Це посередник між кодом та чат-ботом. Його роботу можна простежити на рис. 2.

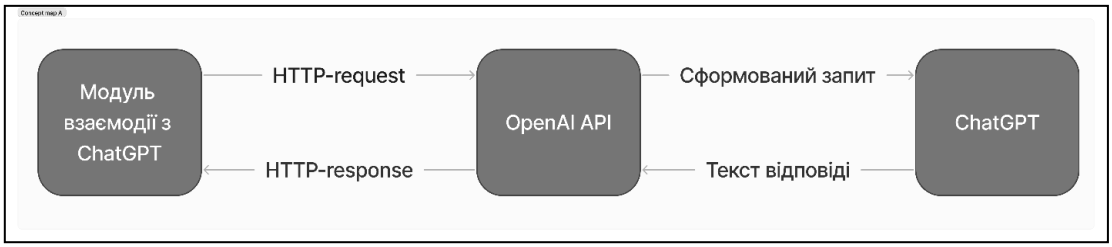


Рисунок 2 – Взаємодія з ChatGPT через OpenAI API

Також до цієї логіки належить текстовий запит, від якого залежить більша частина результату тестових питань. Пройшло 52 ітерації перед тим, як запит

надав найякісніший результат. Під час кожної ітерацій у запит вносилися зміни, щоб отримати все кращі і релевантніші питання.

Далі були об'єднані дві вищеописані частини програми, що дало змогу зберігати тести від різних викладачів різних факультетів та кафедр, а також відповіді на ці питання від студентів. Це відкрило шлях до аналізу даних, який може надати висновки про навчання студентів та покращити його.

Результати показали, що розроблена система ефективно генерує тестові питання з мінімальною кількістю помилок. Питання відповідають навчальній програмі та підходять для оцінювання знань на різних етапах навчання. Система автоматично адаптує складність питань відповідно до рівня студентів, що підвищує точність оцінювання. Тестування підтвердило, що автоматизація створення тестів значно скорочує час підготовки матеріалів, даючи змогу викладачам зосередитися на інших аспектах.

Список використаних джерел

1. Антонов Ю. С., Космінська О. М. Методика аналізу тестових завдань на основі отриманих результатів тестування. *Інформаційні технології і засоби навчання.* 2009. № 4. URL: https://journal.iitta.gov.ua/index.php/itlt/article/view/81/67 (дата звертання: 31.10.2024).

2. Antonov Y., Smoktii K. Expert systems using for answers analysis in automated knowledge control systems. *Herald of Khmelnytskyi National University. Technical Sciences.* Vol. 339(4). P. 323–331. URL: https://heraldts.khmnu.edu.ua/index.php/heraldts/article/view/368/367 (дата звертання: 31.10.2024).

3. Антонов Ю. С. Оцінка повноти відповідей в автоматизованих системах контролю знань. *Наукові праці Донецького національного технічного університету, Серія Інформатика, кібернетика та обчислювальна техніка.* 2012. № 15. С. 113–117.

UDC 519.832:519.863]:338.138(043.2)

Sapozhnikova V., undergraduate student of Computer Science,
Khmelivskyi Y., research assistant at the Department of Information Technologies

ADVERTISING STRATEGY: A GAME THEORY APPROACH

Vasyl' Stus Donetsk National University, Vinnytsia, Ukraine

Determining just when and where to place ads in today's modern, highly competitive landscape in order to maximize impact and return on investment poses a difficult choice for companies. As audiences have continued to fragment across multiple platforms and time slots, how media advertising is placed has become increasingly critical amidst such intense competition for attention [1]. This paper aims to explore how game theory can provide insight into the rival firms' strategic dynamics of advertisement timing and arrive at an optimal decision in this respect.

The issue of timing can hardly be overestimated in this realm, considering that companies operating within today's digital platform-and-traditional-media-laden world must closely consider not only their own optimal timing but also the strategic

choices of their competitors. When multiple firms compete for the same segments of audiences, their advertisements intersect and cause less efficient advertising because of the audience's fatigue and diverted attention. The consequence, therefore, tends to strategic interdependence of decisions within the competitors, which makes game theory an ideal vehicle of analysis [1]. The work offers a close-up view of a basic game-theoretic model of advertisement timing strategies by two competing companies. Analyzing this simplified example, yet some basic lessons can be derived on how strategic interaction in advertising timing decisions works, while simultaneously developing some practical guidelines for companies facing such decisions.

Game theory offers a systematic approach to studying the interactions of multiple decision-makers whose conscious choices affect one another's payoffs. Within the timing of advertising, two critical concepts operate.

First, there is a Nash Equilibrium: given the player's decision depending on the other players, nobody can be better off by unilaterally changing his or her move.

Then, Zero-Sum vs. Non-Zero-Sum Games: Although many consider traditional advertising competitions to be zero-sum games – a company's gain is another one's loss – the actuality of advertisement timing often puts forward non-zero-sum aspects whereby firms achieve mutual benefit through strategic coordination.

Consider two firms, A and B, operating in the same marker with similar products or services. Each firm must choose one of two major timing strategies:

1. Peak Hours (P): This is when the audience is most engaged.

2. Off-Peak Hours (O): This is when the audience are fewer but still remarkably engaged, or at least much higher than during other times.

Key assumptions of this model are that:

- Both firms have a fixed advertising budget.
- The audience's attention is limited, which could be diluted by the presence of many competing ads.
- Companies aim at maximizing the effectiveness of their advertisement, defined as audience reach and engagement.
- The effectiveness of an ad will depend on both timing choice and competitor behavior. Numbers represent relative effectiveness scores on a scale from 1 to 10.

The following payoff matrix illustrates the outcomes for both companies under different timing combinations.

Table 1 – Payoff matrix

Company A / **Company B**	**Peak**	**Off-peak**
Peak	(5,5)	(8,6)
Off-peak	(6,8)	(7,7)

Numbers represent relative effectiveness scores on a scale from 1 to 10. The reduction to 5 for simultaneous peak-hour advertising reflects estimated losses from direct competition for audience attention. The asymmetric 8,6 scores represent the trade-off between peak audience access and competitive interference, while the 7,7

off-peak equilibrium balances smaller audience size against clearer messaging opportunities.

By analyzing this payoff matrix, several key insights arise. They achieve only a moderate level of success when both companies choose to advertise during peak hours since their dilution of audience attention reduces the attention any one company can command. It is represented in our model by having both companies get a score of 5 out of 10 in the above case. The messages from one company fight for viewer mindshare and interfere with messages from the other company.

The more interesting dynamics come up when companies employ different timing strategies. In these cases, both firms can achieve far greater payoffs-the peak hours company reaches an 8 while the off-peak company reaches a 6, demonstrating that such avoidance of direct temporal competition can be beneficial to both, even when gains are not equal. Most interestingly perhaps, the Nash equilibrium in this scenario occurs when both companies choose off-peak hours with scores of 7 for each firm. Although this is perhaps not the theoretical optimum for either firm, it does constitute a steady state that is superior to relentless competition in prime time. This equilibrium point implies that firms may naturally converge toward temporal differentiation in their advertising strategy even in the absence of overt coordination.

Our analysis yields a number of key strategic implications for advertising strategy. Here, one of the most interesting implications brings forth the principle of competitive avoidance: the efficiency of advertising by a firm can be enhanced by selecting time slots that differ from those of its competitors. Direct coordination of firms would involve several legal and practical difficulties, but the market mechanisms would imply, naturally enough, a sort of implicit coordination where consistent patterns of timing would emerge in a self-organizing fashion among competitors.

Of course, these theoretical implications are reflected in the actual advertising practices of the real world. For example, large beverage companies hold key temporal positioning in television advertising by intentionally staggering their advertising schedules to catch prime time for maximum impact with minimum direct competition [2]. In fact, similar patterns are followed in the online world, where companies work to use advanced data analytics in finding those moments in which the activity of their competitors on social media platforms is low and then capitalize on those moments [2].

From this analysis, it is possible to identify a few concrete recommendations that an advertising manager might actually follow in an effort to optimize a timing strategy. This path to success begins with observing competitor timing patterns for potentially revealing openings through which one can effectively differentially position a brand in the marketplace. The traditional appeal of large audience sizes versus the intensity of competition in those time slots can often lead a manager to conclude that less competitive periods may yield better overall returns. This, in turn, needs the elaboration of flexible timing strategies able to respond with a sharp twist to changes in competitors' behavior. Thereafter, sophisticated data analytics should be made use of to unmask those underused time slots which may yield a better return on investment despite probably smaller audience sizes [2]. In sum, these practices form a comprehensive approach to strategic timing in advertising-a way of balancing theoretical insights with practical market realities.

Game theory provides a vehicle for examining the strategic dynamics of advertisement timing. The analysis in this paper strongly suggests that, in many circumstances, firms can do better by taking competitors into account and seeking timing strategies that trade audience's reach against competitive intensity. The Nash equilibrium thus detected suggests that companies may be inherently pulled towards timing patterns that avoid direct competition, absent of any overt coordination.

While this analysis has several valuable insights, several limitations should be noted:

1. Real-world advertising markets may be too richly detailed to be usefully approximated by this two-player, two-strategy model.

2. The assumption of fixed advertising budgets is somewhat artificial and does not accurately model the way marketing resources are allocated.

3. The model does not consider, for instance, issues of audience segmentation or platform-specific timing.

The limitation notwithstanding, the game-theoretic framework here provides the basic vertices on which a brilliant comprehension and optimization of advertisement timing decisions stand in competitive markets.

Список використаних джерел

1. Mastering Meta: A Comprehensive Guide to Best Practices in Advertising for Beginners. *PB&J Promotions LLC.* 2023. URL: https://pbjmarketing.com/blog/mastering-meta-a-comprehensive-guide-to-best-practices-in-advertising-for-beginners

2. Dominici G. Game theory as a marketing tool: uses and limitations. *Elixir Marketing.* Vol. 36 P. 3524–3528. URL: https://www.academia.edu/1084466/Game_theory_as_a_marketing_tool_uses_and_limitations

УДК 004.738.5:339.138

Козачок А. О., здобувачка вищої освіти,
Хмелівський Ю. С., асистент кафедри
інформаційних технологій

ЗАСТОСУВАННЯ СИСТЕМИ РЕКОМЕНДАЦІЙ У ОНЛАЙН-МАГАЗИНІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Зі стрімким розвитком електронної комерції зростає конкуренція між онлайн-магазинами, що змушує їх шукати інноваційні способи утримання клієнтів та збільшення продажів. Одним із таких способів є впровадження систем рекомендацій. Рекомендаційні системи не лише допомагають споживачам швидше знаходити потрібні товари, але й персоналізують взаємодію клієнта з платформою. Успіх гігантів, як-от Amazon чи Netflix, значною мірою пов'язаний із використанням цих технологій.

Система рекомендацій базується на аналізі великих обсягів даних про користувачів, товари та їхню взаємодію. Для цього використовуються такі методи:

колаборативна фільтрація, контентна фільтрація та гібридні системи. Далі детальніше про кожну.

Першою розглянемо колаборативну фільтрацію. Основна її ідея полягає у використанні поведінкових схожостей між користувачами. Наприклад, якщо два клієнти мають схожий набір вподобань (тобто купували подібні товари), то система може запропонувати товари одного з них іншому. Формально схожість між двома користувачами можна обчислити за допомогою косинусної подібності:

$$\sin(u, v) = \frac{\sum_{i=1}^{n} r_{u,i} \cdot r_{u,i}}{\sqrt{\sum_{i=1}^{n} r^2{}_{u,i}} \cdot \sqrt{\sum_{i=1}^{n} r^2{}_{u,i}}},$$

де $r_{u,i}$ – оцінки користувачів u, v для товару i.

Наступна – контентна фільтрація. Фокусується на властивостях самих товарів і вподобаннях конкретного користувача. Наприклад, якщо клієнт часто купує книги певного жанру, система рекомендуватиме схожі за жанром товари. Рейтинг прогнозується за такою формулою:

$$\hat{r}_{u,i} = \sum_{k=1}^{m} \omega_k \cdot x_{i,k},$$

де $x_{i,k}$ – характеристика товару «i» (наприклад, жанр, ціна), а ω_k – вага відповідної характеристики.

І останнє – це гібридні системи. Ці підходи комбінують переваги колаборативної та контентної фільтрації. Наприклад, алгоритм може використовувати дані про історію покупок клієнтів та властивості товарів для досягнення більшої точності.

Рекомендаційні системи дають змогу вирішувати кілька завдань:

- Збільшення середнього чека досягається завдяки пропозиціям супутніх або додаткових товарів. Наприклад, якщо клієнт купує смартфон, система може запропонувати захисний чохол чи навушники.
- Кожен клієнт отримує унікальні рекомендації, про які відповідають його вподобанням. Це збільшує шанси на покупку.
- Система може порекомендувати товари, які клієнт не знав, але вони відповідають його інтересам.

Розглянемо приклад із реального життя: платформа Amazon використовує складну гібридну модель, яка базується на багаторівневій аналітиці, включно з аналізом історії покупок, уподобань користувачів, демографічних даних і сезонних трендів.

Попри численні переваги, є кілька проблем, з якими стикаються онлайн-магазини під час впровадження рекомендаційних систем:

1. Збір і зберігання даних, тому що для коректної роботи алгоритмів необхідно зібрати велику кількість даних про користувачів, їхні вподобання та історію покупок. Це може спричинити складнощі із забезпеченням конфіденційності.

2. Для аналізу великих даних потрібні потужні сервери та ефективні алгоритми оптимізації.

3. Якщо платформа пропонує широкий асортимент товарів (тисячі чи навіть мільйони позицій), виникає складність у створенні релевантних рекомендацій для кожного клієнта.

Отже, системи рекомендацій продовжують розвиватися. Наприклад, сучасні дослідження зосереджуються на інтеграції нейронних мереж та моделей глибокого навчання. Алгоритми на основі рекурентних нейронних мереж (RNN) або трансформерів дають змогу прогнозувати вподобання клієнтів з урахуванням контексту та часу. До того ж активно розробляються пояснювані рекомендаційні системи (explainable AI), які пояснюють клієнту, чому той чи інший товар йому пропонується.

Можна зауважити, що системи рекомендацій є невід'ємною частиною сучасних онлайн-магазинів, які прагнуть персоналізувати взаємодію з клієнтами та збільшити продажі. Попри наявні виклики, розвиток технологій обробки даних і машинного навчання робить ці системи дедалі ефективнішими. Майбутнє цих технологій відкриває нові горизонти для бізнесу, покращуючи досвід споживачів і сприяючи росту компаній.

Список використаних джерел

1. Дослідження методів побудови рекомендаційних систем в мережі Інтернет / Є. В. Мелешко та ін. *Збірник наукових праць «Системи управління, навігації та зв'язку».* Полтавський національний технічний університет ім. Ю. Кондратюка. 2018. Т. 1, № 47. С. 131–136.

2. Лобур М. В., Шварц М. Є., Стех Ю. В. Моделі і методи прогнозування рекомендацій для колаборативних рекомендаційних систем. *Вісник НУ «Львівська політехніка». Серія: Інформаційні системи та мережі.* 2018. № 901. Львів: Видавництво Львівської політехніки. С. 68–75.

3. Моделі формування рекомендацій у інтелектуальних системах електронної комерції / О. Ю. Чередніченко, О. В. Янголенко, О. В. Іващенко, О. М. Матвєєв. *Системи обробки інформації*. 2020. № 1(160). С. 32–39.

4. Клюйко Г. Товарні рекомендації для ecommerce на сайті та в листах. 2021. URL: https://esputnik.com/uk/blog/tovarni-rekomendaciyi-dlya-ecommerce-na-sajti-ta-v-listah

УДК 004.422.5

Мигун Р. С., здобувач вищої освіти,
Потапова Н. А., канд. екон. наук, доцент,
доцент кафедри інформаційних технологій

ЗАСТОСУВАННЯ АЛГОРИТМУ ПОБУДОВАНОЇ ЦИФРОВОЇ ПІДПИСКИ З ВИКОРИСТАННЯМ ЕЛІПТИЧНИХ КРИВИХ (ECDSA) У БЛОКЧЕЙН-ТЕХНОЛОГІЯХ

Донецький національний університет імені Василя Стуса, м. Вінниця

ECDSA (абревіатура від *Elliptic Curves Digital Signature Algorithm*, алгоритм побудованої цифрової підписки з використанням еліптичних кривих) –

це схема криптографії на основі еліптичних кривих (Elliptic Curve Cryptography або ECC).

ECDSA використовується в багатьох системах безпеки, зокрема для захисту повідомлень і в основі безпеки Bitcoin. До того ж він застосовується для забезпечення безпеки транспортного рівня (TLS), що замінив SSL, шляхом шифрування з'єднань між веббраузерами та вебпрограмами. HTTPS-з'єднання, позначене іконкою замка у браузері, також захищається за допомогою підписаних сертифікатів через ECDSA.

Еліптична криптографія (ECC) має суттєву перевагу перед RSA завдяки меншому розміру ключів за умови однакового рівня безпеки. Наприклад, ECC з ключем 256 біт забезпечує таку ж криптографічну стійкість, як RSA з ключем 3 072 біт. Це робить ECC більш ефективною з погляду використання пам'яті, обчислювальних ресурсів і пропускної здатності, що є критично важливим для блокчейн-екосистем.

Ключовими компонентами ECDSA є:

1) Генерація пари ключів: це передбачає вибір випадкового закритого ключа, який є секретним числом, і використання його для отримання відкритого ключа, яким можна поділитися публічно. Відкритий ключ виходить шляхом множення фіксованої точки на еліптичній кривій на закритий ключ.

2) Хешування повідомлення: повідомлення, яке потрібно підписати, хешується за допомогою криптографічної хеш-функції, яка створює хеш-значення фіксованої довжини.

3) Підписання повідомлення. Цифровий підпис ECDSA створюється шляхом виконання ряду математичних операцій над хеш-значенням і закритим ключем. Результатом цих операцій є унікальний підпис, який може бути згенерований лише закритим ключем відправника.

4) Перевірка підпису: одержувач повідомлення використовує відкритий ключ для перевірки підпису. Якщо підпис дійсний, одержувач може бути впевнений, що повідомлення надіслано власником закритого ключа.

5) Параметри домену: еліптична крива E, визначена в дискретному просторі Fq з характеристикою p і параметрами домену базової точки G, може бути розподілена групою об'єктів або унікальною для одного користувача.

Безпека ECDSA залежить від складності розв'язання проблеми дискретного логарифмування еліптичної кривої, яка передбачає пошук закритого ключа за відкритим ключем і точкою генератора. Вважається, що проблема дискретного логарифмування еліптичної кривої обчислювально неможлива для достатньо великих розмірів ключів, що робить ECDSA надійним алгоритмом цифрового підпису.

Під час роботи автори вирішили реалізувати класи PrivateKey та PublicKey на мові програмування C#. У випадку з PrivateKey – ми маємо клас, який містить у собі криву та секретне число і може підписувати повідомлення:

```
public class PrivateKey
{
    private Curve Curve { get; }
    private BigInteger Secret { get; }
```

```
    public PrivateKey(Curve curve, BigInteger secret)
    {
        this.Curve = curve;
        this.Secret = secret;
    }

    public PublicKey PublicKey()
    {
        var publicPoint = MathHelper.Multiply(this.Curve.G, this.Secret, this.Curve.N, this.Curve.A, this.Curve.P);
        return new PublicKey(publicPoint, this.Curve);
    }

    public Signature Sign(string message)
    {
        var messageBytes = Encoding.UTF8.GetBytes(message);
        var hashBytes = SHA256.HashData(messageBytes);
        var numberMessage = new BigInteger(hashBytes.Reverse().ToArray());
        var curve = this.Curve;

        BigInteger r;
        BigInteger s;
        Point randSignPoint;
        do
        {
            var randNum = BigIntegerRandom.Between(1, curve.N - 1);
            randSignPoint = MathHelper.Multiply(curve.G, randNum, curve.N, curve.A, curve.P);
            r = randSignPoint.X % curve.N;
            s = ((numberMessage + r * this.Secret) * (MathHelper.Inv(randNum, curve.N))) % curve.N;
        } while (r == 0 || s == 0);

        int recoveryId = (int)(randSignPoint.Y & 1);
        if (randSignPoint.Y > curve.N)
        {
            recoveryId += 2;
        }

        return new Signature(r, s, recoveryId);
    }
}
```

Тепер перейдемо до класу PublicKey, який є парою до PrivateKey і може перевіряти, чи було підписано це повідомлення конкретно цим приватним ключем.

```
public class PublicKey
{
    public Point Point { get; }
    public Curve Curve { get; }

    public PublicKey(Point point, Curve curve)
    {
        this.Point = point;
        this.Curve = curve;
    }

   public bool Verify(string message, Signature signature)
    {
        var messageBytes = Encoding.UTF8.GetBytes(message);
        var hashBytes = SHA256.HashData(messageBytes);
        var numberMessage = new BigInteger(hashBytes.Reverse().ToArray());
        var curve = this.Curve;
```

```
        if (signature.R < 1 || signature.R > curve.N - 1)
        {
            return false;
        }

        if (signature.S < 1 || signature.S > curve.N - 1)
        {
            return false;
        }

        var w = MathHelper.Inv(signature.S, curve.N);
        var u1 = (numberMessage * w) % curve.N;
        var u2 = (signature.R * w) % curve.N;
        var point = MathHelper.Add(MathHelper.Multiply(curve.G, u1, curve.N, curve.A, curve.P),
MathHelper.Multiply(this.Point, u2, curve.N, curve.A, curve.P), curve.A, curve.P);

        if (point.X == 0 && point.Y == 0)
        {
            return false;
        }

        return point.X % curve.N == signature.R;
    }
}
```

Отже, ECDSA забезпечує високу криптографічну безпеку для блокчейн-транзакцій, використовуючи еліптичні криві. Реалізація класів PrivateKey і PublicKey на C# дає змогу ефективно підписувати та перевіряти повідомлення, що є основою для безпеки у блокчейн-мережах.

Список використаних джерел

1. Elliptic Curve Digital Signature Algorithm. *HYPR.* URL: https://surl.li/lxxejp (дата звернення: 20.12.2024).

2. Raza S. Ethereum’s Elliptic Curve Digital Signature Algorithm (ECDSA). *Medium.* 2023. URL: https://fitsaleem.medium.com/ethereums-elliptic-curve-digital-signature-algorithm-ecdsa-88e1659f4879

УДК 004.415.28

Перепелиця А. С., здобувач вищої освіти,
Бабаков Р. М., д-р техн. наук, професор
кафедри інформаційних технологій

ІНФОРМАЦІЙНА ПЛАТФОРМА ДЛЯ АНАЛІЗУ НАВЧАННЯ ТА КАР’ЄРНИХ МОЖЛИВОСТЕЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

Проблема вибору майбутньої професійної траєкторії є актуальною для старшокласників, які часто стикаються з невизначеністю, зумовленою недостатньою поінформованістю про їхні здібності, інтереси та можливості на ринку праці [1, 2]. Для розв’язання цієї проблеми було розроблено інформаційну

систему, яка узагальнює результати навчання учнів і надає персоналізовані рекомендації щодо вибору спеціальності для вступу в університет.

Сучасна освіта активно інтегрує інформаційні технології [3], які дають змогу автоматизувати аналіз даних і підтримувати прийняття рішень. Унікальність запропонованої системи полягає у поєднанні функцій журналювання академічних досягнень із базовою профорієнтацією, заснованою на результатах психометричного аналізу. Це сприяє більш усвідомленому вибору професійного шляху, забезпечує комплексний аналіз особистості та уникає суб'єктивності у профорієнтаційних рекомендаціях, яка може виникнути через людський фактор.

Основні завдання такої платформи [4]:

1) Розробити інструментарій для збору та обробки навчальних даних і психометричного тестування.

2) Інтегрувати моделі штучного інтелекту для формування персоналізованих рекомендацій.

3) Забезпечити доступ до системи для учнів, вчителів і батьків із різними рівнями прав доступу.

4) Реалізувати функції звітування, які допоможуть надавати рекомендації у зручному форматі.

Розроблена платформа базується на таких технологіях:

- Бекенд реалізований на Kotlin із використанням фреймворка Spring Boot. Spring Boot є потужним і функціональним фреймворком, що забезпечує стабільність роботи системи, а модуль Spring Security забезпечує механізми автентифікації, авторизації та захисту від атак, як-от CSRF, SQL-ін'єкції тощо.
- Фронтенд створено за допомогою JavaScript і бібліотеки React, що гарантує адаптивність і зручність користувацького інтерфейсу.
- Для зберігання даних використано об'єктно-реляційну СУБД PostgreSQL, яка відповідає вимогам до масштабованості й обробки великих обсягів даних.
- Модель штучного інтелекту, що використовується для профорієнтаційних рекомендацій, реалізована із застосуванням платформи H2O.ai. Це дає змогу аналізувати навчальні досягнення у поєднанні з результатами психологічного тестування та, за потреби, надає можливість модифікувати модель ШІ, аби надалі за наявності більшої кількості даних давати більш точні передбачення, базуючись на реальних кейсах вступу випускників.

Наразі на платформі реалізовано такий функціонал:

1) Централізоване збереження даних про успішність учнів і результати психометричних тестів.

2) Генерація профорієнтаційних рекомендацій на основі аналізу даних. Зокрема, враховується весь перелік навчальних предметів, затверджений Міністерством освіти і науки України, і на платформі є можливість пройти психологічні тести для психометричного аналізу. Наразі для проходження доступні тести Айзенка та Голланда [5].

3) Кросплатформений доступ до інформації для учнів, вчителів і батьків через багаторівневу систему авторизації.

4) Система видів роботи, що дає можливість у межах одного уроку оцінити роботу учня за різними видами активностей, не переносячи оцінки на клітинки інших днів (як це виглядає для вчителя, показано на рис. 1).

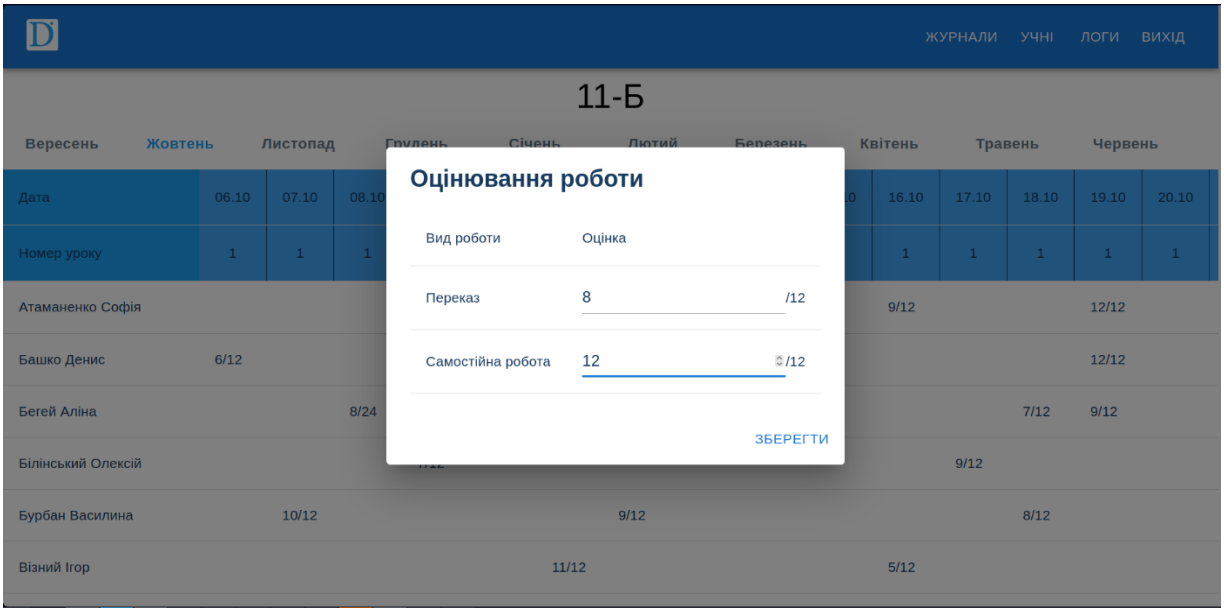


Рисунок 1 – Інтерфейс редагування кількох оцінок у межах одного уроку

Розроблена система допомагає:

1) знизити рівень стресу та невизначеності серед учнів під час вибору професійної траєкторії;

2) надати учням обґрунтовані рекомендації, які сприятимуть їхньому професійному та особистісному розвитку;

3) забезпечити вчителям і батькам зручні інструменти для моніторингу навчального прогресу й підтримки учнів у їхніх рішеннях.

Інформаційна система для аналізу навчання та кар'єрних можливостей є інноваційним рішенням, що об'єднує автоматизацію освітнього процесу з профорієнтаційною підтримкою. Її впровадження дасть змогу підвищити якість освітнього процесу, сприяти усвідомленому вибору професії та поліпшити підготовку учнів до викликів ринку праці.

Список використаних джерел

1. Sihite J. The student preference to choose higher education: a case study in west Jakarta 2019. *European journal of business and mangement.* 2019. № 25. С. 59–64. URL: https://core.ac.uk/download/pdf/276531308.pdf (дата звернення: 20.12.2024).

2. Шкіль Л. Тільки кожен шостий студент закінчує IT-курси до кінця – дослідження. *AIN.UA.* URL: https://ain.ua/2023/01/05/tilky-kozhen-shostyj-student-zakinchuye-it-kursy-do-kinczya-doslidzhennya/ (дата звернення: 20.12.2024).

3. Електронні класні журнали та щоденники з можливостями дистанційного навчання. *Нові Знання.* URL: https://nz.ua (дата звернення: 20.12.2024).

4. Перепелиця А. С., Бабаков Р. М. Інформаційна система для узагальнення результатів навчання з функцією профорієнтації. *Прикладні аспекти сучасних міждисциплінарних досліджень:* Всеукр. науково-практ. конф., (м. Вінниця, 1 листоп. 2024 р.). Вінниця, 2024. URL: https://jpasmd.donnu.edu.ua (дата звернення: 20.12.2024).

5. Опитувальник професійної спрямованості (ОПС) Д. Голанда. *Тернопільський обласний центр зайнятості.* URL: https://ter.dcz.gov.ua/publikaciya/opytuvalnyk-profesiynoyi-spryamovanosti-ops-d-golanda (дата звернення: 20.12.2024).

УДК 004.056:004.052;004.652

Плець В. В., здобувач вищої освіти,
Потапова Н. А., канд. екон. наук., доцент,
доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ ХЕШ-ФУНКЦІЙ І ХЕШ-ТАБЛИЦЬ У КІБЕРБЕЗПЕЦІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Хеш-функція – це математичний алгоритм, який перетворює дані довільної довжини (текст, числа, файли тощо) у хеш-значення (також відоме як хеш або дайджест) фіксованого розміру. Незалежно від розміру вхідних даних, результатом роботи хеш-функції завжди буде рядок однакової довжини, що робить її корисною для багатьох завдань, як-от:

1. Перевірка цілісності даних: хешування використовується для створення контрольних сум, які дають змогу перевіряти, чи не було змінено дані під час передачі або зберігання.
2. Захист паролів: у базах даних замість збереження паролів зберігають їх хеші, що підвищує безпеку.
3. Цифрові підписи та сертифікати: хеш-функції використовуються в криптографії для створення підписів і перевірки автентичності даних.
4. Оптимізація пошуку: у структурах даних, як-от хеш-таблиці, хешування допомагає швидко знаходити потрібні елементи.

Хороша хеш-функція повинна відповідати таким властивостям: детермінованість (однакові вхідні дані завжди дають один і той самий хеш), швидкість (обчислення хешу має бути ефективним), стійкість до колізій (різні вхідні дані повинні давати різні хеш-значення), невідворотність (за хеш-значенням має бути практично неможливо відновити початкові дані).

Популярні хеш-функції включають MD5, SHA-1, SHA-256 та інші. Однак деякі старі функції, як-от MD5 і SHA-1, більше не вважаються безпечними через вразливість до колізій [1]. На рис. 1 та 2 наведено обчислення хешу на C# (SHA-256) та результат виконання програмного коду.

Хеш-таблиці та хеш-функції відіграють ключову роль у сучасних системах кібербезпеки завдяки своїй здатності ефективно обробляти великі обсяги даних та забезпечувати захист від несанкціонованого доступу. Хеш-функції використовуються для створення унікальних і незворотних значень, що допомагають забезпечити цілісність і автентичність даних, а також створюють основи для алгоритмів підпису та перевірки повідомлень. Натомість хеш-таблиці є важливим інструментом для швидкого пошуку та перевірки великих наборів даних, як-от списки заблокованих IP-адрес чи паролів, що підвищує ефективність кіберзахисту.

```csharp
using System;
using System.Security.Cryptography;
using System.Text;

class Program
{
    static void Main()
    {
        string data = "Example data";

        // Перетворення рядка у байти
        byte[] dataBytes = Encoding.UTF8.GetBytes(data);

        // Створення об'єкта SHA256
        using (SHA256 sha256 = SHA256.Create())
        {
            // Обчислення хешу
            byte[] hashBytes = sha256.ComputeHash(dataBytes);

            // Перетворення байтів у шістнадцятковий рядок
            StringBuilder hashString = new StringBuilder();
            foreach (byte b in hashBytes)
            {
                hashString.Append(b.ToString("x2"));
            }

            Console.WriteLine("Хеш: " + hashString);
        }
    }
}
```

Рисунок 1 – Код для обчислення хешу на C#

```
Хеш: e01757d4eae002c09e8e033373d2ea288452d6184cc3344086090972f029fa7e
```

Рисунок 2 – Результат коду

Хеш-таблиці – це структура даних для зберігання та швидкого пошуку інформації, наприклад, чорних списків IP-адрес чи інших загроз [4].

Застосування хеш-таблиць і хеш-функцій дає змогу значно знизити ризики кібератак, як-от маніпуляції з даними або спроби злому паролів. Однак важливо враховувати, що успішне використання цих технологій залежить від правильної реалізації та вибору хеш-функцій з високим рівнем безпеки, як-от SHA-256 чи інші алгоритми, що протистоять сучасним методам злому, зокрема атакам з використанням колізій.

```csharp
using System;
using System.Collections.Generic;

class Program
{
    static void Main(string[] args)
    {
        // Створення списку заблокованих IP
        HashSet<string> blacklist = new HashSet<string>
        {
            "192.168.1.1",
            "203.0.113.5",
            "10.0.0.2"
        };

        // IP-адреса для перевірки
        string ipToCheck = "203.0.113.5";

        if (IsIpBlacklisted(blacklist, ipToCheck))
        {
            Console.WriteLine($"IP {ipToCheck} заблоковано.");
        }
        else
        {
            Console.WriteLine($"IP {ipToCheck} дозволено.");
        }
    }

    static bool IsIpBlacklisted(HashSet<string> blacklist, string ip)
    {
        return blacklist.Contains(ip);
    }
}
```

Рисунок 3 – Код використання хеш-таблиці для чорного списку IP

```
IP 203.0.113.5 заблоковано.
```

Рисунок 4 – Результат коду використання хеш-таблиці для чорного списку IP

Хеш-функції застосовуються у криптографічних протоколах, як-от TLS, для забезпечення конфіденційності й автентичності [5].

```
using System;
using System.Security.Cryptography;

class Program
{
    static void Main(string[] args)
    {
        string sessionHash = GenerateSessionHash();
        Console.WriteLine("Унікальний хеш сеансу: " + sessionHash);
    }

    static string GenerateSessionHash()
    {
        using (var randomNumberGenerator = new RNGCryptoServiceProvider())
        {
            byte[] tokenBytes = new byte[32];
            randomNumberGenerator.GetBytes(tokenBytes);

            using (SHA256 sha256 = SHA256.Create())
            {
                byte[] hashBytes = sha256.ComputeHash(tokenBytes);
                return BitConverter.ToString(hashBytes).Replace("-", "").ToLower();
            }
        }
    }
}
```

Рисунок 5 – Код для генерації унікального хешу для сеансу

```
Унікальний хеш сеансу: 84c29f5f5424e9ccd5972ec34ea273146cb886efb40572d4c5fc6a4c68ae162b
```

Рисунок 6 – Результат коду для генерації унікального хешу для сеансу

Перспективними напрямами використання хеш-функцій та хеш-таблиць є: постквантова криптографія (розробка хеш-функцій, стійких до атак квантових комп’ютерів), а також блокчейн (забезпечення прозорості та незмінності транзакцій за допомогою хеш-функцій).

Отже, хеш-таблиці та хеш-функції є важливими елементами інфраструктури кібербезпеки, що забезпечують захист даних, швидкий доступ до критичних відомостей і підтримку високої надійності в умовах постійно змінюваного кіберпростору.

Список використаних джерел

1. Blog.whitebit. URL: https://blog.whitebit.com/uk/what-is-a-hash-in-blockchain/
2. Thekernel.ua. URL: https://thekernel.ua/identyfikatsiia-avtentyfikatsiia-ta-avtoryzatsiia/
3. support.microsoft.com. URL: https://support.microsoft.com/ukua/office/%D0%B4%D0%BE%D0%B4%D0%B0%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%B0%D0%B1%D0%BE
4. Vpnunlimited.com. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/hashing?srsltid= AfmBOopWKyHmylsoS3Bf0VueGMLDPIIIT8oKC-4ZUTvOHq9gwXs2UB6C
5. Vpnunlimited.com. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/hashing?srsltid=AfmBOopWKyHmylsoS3Bf0VueGMLDPIIIT8oKC-4ZUTvOHq9gwXs2UB6C

УДК 681.5.015:007

Русавський О. О., аспірант,
Роштейн О. П., д-р техн. наук,
професор кафедри інформаційних технологій

ІДЕНТИФІКАЦІЯ НЕЛІНІЙНИХ ЗАЛЕЖНОСТЕЙ НЕЧІТКИМИ КОГНІТИВНИМИ КАРТАМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. У тезі запропоновано метод моделювання на основі нечітких когнітивних карт (НКК), який можна розглядати як альтернативу багатовимірному регресійному аналізу. Завдяки такому аналізу система буде здатна вилучати зв'язки між вхідними даними (концептами), отриманими під час експерименту. Окрім вхідних параметрів, система опрацьовуватиме ваги дуг як невідомі параметрами, які оцінюються у два етапи.

1. Оцінка на основі приблизної близькості значення стовпців таблиці спостережень даних входу-виходу.

2. Використання таблиці спостережень для коригування ваг за допомогою методу найменших квадратів та оптимізації за допомогою генетичного алгоритму.

Метод можна застосовувати для обробки даних у галузях, як-от медична, технічна діагностика, прогнозування та сценарне моделювання.

Ідентифікація – це процес побудови математичної моделі, яка визначає взаємозв'язок між концептами та експериментальними даними (Эйкхофф, 1974; Ципкін, 1984; Ljung,1999). Зазвичай задачу ідентифікації вирішують у два етапи. На першому етапі відбувається «груба» побудова моделі, що апроксимує зв'язки між концептами і корегується параметрами налаштування. На другому етапі йде процес підбору параметрів, що мінімізує відстань між модельними та експериментальними даними. Ідентифікація використовується для опису параметрів і структур математичної моделі, які найкраще описують поведінку реальної системи. Цей етап є ключовим для створення точних і коректних моделей, які можуть бути використані для прогнозування, аналізу і прийняття рішень.

Основи теорії ідентифікації були закладені в роботах Н. Вінера, А. Н. Колмогорова та Р. Кальмана.

Об'єктом дослідження є нелінійні залежності зі входами та виходами в медицині, економіці, бізнесі та інших галузях.

Предметом дослідження є ідентифікація нелінійних залежностей на основі спостережень, як-от експериментальні дані або експертні оцінки, за допомогою НКК.

Метою роботи є розробка взаємопов'язаного комплексу математичних моделей, алгоритмів і програмного забезпечення для ідентифікації нелінійних залежностей.

Наукова новизна методу буде визначатися перевагою цього методу, порівняно з наявними аналогами, як-от методи ідентифікації на основі нечітких правил і відношень [1].

Практична цінність роботи полягає у створенні експертно-моделюючої системи, яку можна використовувати для розв'язання задач ідентифікації в різноманітних задачах обробки даних для прогнозування, діагностики та прийняття рішень.

Методами дослідження є нечітка логіка, генетичні алгоритми, нейронні мережі та комп'ютерний експеримент.

Останні дослідження з цієї тематики перераховані в багатьох відомих наукометричних базах даних, де можна знайти застосування окремо нечітких когнітивних карт для вирішення задач моделювання комплексних систем керування або модулів таких систем, наприклад, у статті «A generalised fuzzy cognitive mapping approach for modelling complex systems» [2].

Вибір НКК як інструменту для моделювання дасть можливість застосувати як індуктивний та дедуктивний підхід до моделювання. Якщо концепти інтерпретуються як вхідні та вихідні змінні, то НКК виступатиме індуктивною моделлю. Якщо концепти і зв'язки між ними інтерпретуються як задана структура системи, то НКК виступатиме як модель дедуктивного виведення значень концептів на кожній ітерації моделювання. Один із алгоритмів – це формула покрокової динамічної зміни значень концептів:

$$A_i^{k+1} = A_i^k + \sum_{j=1}^{n} \left(A_j^k - A_j^{k-1}\right) \cdot w_{ji}, \qquad (1)$$

де A_i^{k+1} – це значення концепту в кроці $k+1$;

A_j^k та A_i^k – це значення концептів в кроці k;

w_{ji} – вплив концептів один на одного;

A_j^{k-1} – це значення концепту на попередньому кроці.

Так, наприклад, нечітка когнітивна карта з п'ятьма концептами, де чотири з них – вхідні параметри, що впливають на результат прогнозування, і п'ятий – це відповідно сам результат прогнозування. Приклад такої когнітивної карти видно на рис. 1.

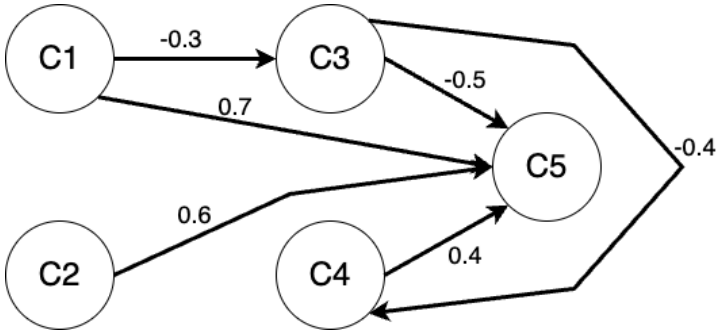


Рисунок 1 – Візуальне представлення нечіткої когнітивної карти із вагами дуг впливів

Щоб вирішити проблему оптимізації, можна використати генетичний алгоритм. Алгоритм працює з популяцією можливих рішень, які називаються

хромосомами. Він імітує процеси еволюції, як-от схрещування, мутація і відбір. Псевдокод такого алгоритму виглядає так:

Початок
Ініціалізувати початкову популяцію P(t)
Оцінити популяцію P(t) за допомогою функції пристосованості

Поки (умови завершення не виконано) **робити**
Виконати операцію схрещування для отримання нових хромосом C(t)
Виконати мутацію хромосом C(t)
Оцінити нову популяцію C(t) за допомогою функції пристосованості
Вибрати нову популяцію P(t+1) на основі P(t) і C(t)
t = t + 1
Кінець поки
Кінець

У випадку ідентифікації нелінійних залежностей хромосомою буде виступати рядок ненульової матриці елементів $W_0 = [w_{ji}]$, $w_{ji} = R\left[\underline{w_{ji}}, \overline{w_{ji}}\right]$, де $R = [\underline{x}, \overline{x}]$ – це операція знаходження випадкового значення рівномірно розподіленого в інтервалі $[\underline{x}, \overline{x}]$; W_0 – це матриця сил впливів дуг.

У цій тезі запропоновано підхід використання НКК як альтернативи багатовимірному регресійному аналізу для вилучення зв'язку концептів з експериментальних даних. Концепти – це вхідні дані, що залежать від взаємодії один з одним завдяки вагам дуг, що визначаються декількома етапами. Пропонується використати генетичний алгоритм для вирішення проблеми оптимізації.

Список використаних джерел

1. Fuzzy cognitive map and mean square method in empirical modeling: Application in economics / A. Rotshtein, A. Yosef, T. Neskorodeva, D. Katielnikov. *Expert Systems with Applications.* 2024. Vol. 247. P. 4–5. DOI: 10.1016/j.eswa.2024.123176.

2. Nair A., Reckien D., Maarseveen M. F. A. M. A generalised fuzzy cognitive mapping approach for modelling complex systems. *Applied Soft Computing.* 2019. Vol. 84. P. 1–2. DOI: 10.1016/j.asoc.2019.105754.

3. Rotshtein A. P., Rakytyanska H. B. Fuzzy Evidence in Identification, *Forecasting and Diagnosis.* Heidelberg: Springer. 2012. Vol. 275. P. 330. DOI: 10.1007/978-3-642-25786-5_1.

УДК 004.94 +519.876.5

Семенюк А. М., здобувач вищої освіти,
Якубич К. О., асистент кафедри
інформаційних технологій

СИСТЕМА ЕКСПЕРТНОЇ ОЦІНКИ НА БАЗІ FUZZY LOGIC

Донецький національний університет імені Василя Стуса, м. Вінниця

У цій роботі розглядається медична система експертного типу, яка дає змогу використовуючи об'єктивну оцінку стану особи, що представлена у вигляді

лінгвістичних змінних множин нечіткої логіки, проводити класифікацію осіб за станом втрати працездатності.

Будь-яка експертна система базується на трьох основних поняттях:

- факти;
- правила;
- керуюча структура (КР).

Факти – це вхідна інформація. У більшості випадків впливу на її отримання, вигляд, правильність експерт не має, тобто працює з «тим, що є».

Правила та керуюча структура – це вже прерогатива розробника та користувача. Правильно сформульовані правила – методи обробки, аналізу, перетворення – дають змогу створити експертну систему, результатом роботи якої є достовірна інформація (так, можливо, не початковій стадії потрібно буде провести навчання «псевдоінтелекту»).

Для отримання якісної моделі необхідно провести аналіз за різними обліковими ознаками, що дасть можливість дослідити кожен елемент множини вхідних даних, і також всю сукупність загалом.

Але оскільки багато медичних параметрів, які описують стан людини, є елементами, які підпадають під поняття нечіткої логіки, то підготовку такої інформації необхідно проводити з використанням моделей нечіткого керування (рис. 1). Ці операції дадуть змогу перейти від якісних до кількісних ознак.

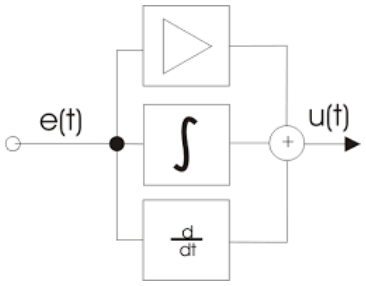


Рисунок 1 – Приклад моделі оцифрування вхідних даних

Отримані дані потрібно згрупувати за спільними рисами. Для цього виконуємо ранжування значень. Якщо вони потрапляють до однієї градації, то всім їм приписують однаковий ранг, який розраховують за формулою:

$$R_n(x) = \sum_{i=0}^{n-1} y_i + \frac{y_n + 1}{2}, \qquad (1)$$

де n – номер градації;

R_n – ранг кожного значення ознаки, що потрапив до градації;

y_n – кількість значень, що потрапили до градації n (y_0 приймається таким, що дорівнює 0).

Для перевірки достовірності ранжування здійснюють такі дії: знаходимо суму всіх рангів і порівнюємо з перевірочною сумою, яку визначаємо за формулою:

$$S_R^T = \frac{N(N+1)}{2}. \quad (2)$$

Для оцінки впливу даних на досліджуваний показник проводиться однофакторний регресійний аналіз залежності від кожної вхідної величини (змінної) та розраховуються коефіцієнти впливу – «ваги» даних.

$$y = \frac{\int_{\underline{y}}^{\overline{y}} y \cdot \mu_{\tilde{y}}(y)dy}{\int_{\underline{y}}^{\overline{y}} \mu_{\tilde{y}}(y)dy}. \quad (3)$$

Керуюча структура по своїй суті – це піраміда прийняття рішень (рис. 2) яка складається з окремих шарів, побудованих на елементарних комірках.

Ці комірки є моделлю аксона (рис. 3) – штучного нейрона (інколи наз. «математичний нейрон») і зазвичай являють собою нелінійну функцію від єдиного аргументу – лінійної комбінації всіх вхідних сигналів. Цю функцію називають функцією активації [2] або функцією спрацьовування, функцією передавання. Отриманий результат посилається на єдиний вихід.

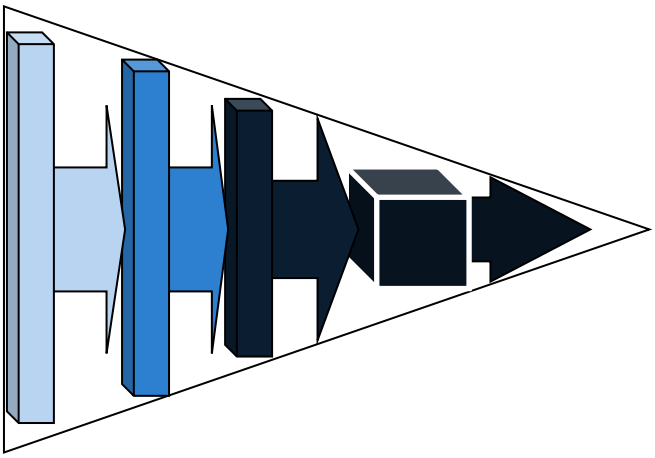


Рисунок 2 – Модель керуючої структури

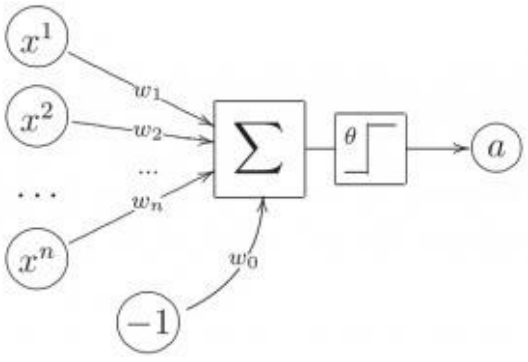


Рисунок 3 – Приклад комірки прийняття рішень (модель аксона)

У шарах штучні нейрони об’єднуються у мережі – перцептрони (рис. 4) [4], коли виходи одних нейронів з’єднують із входами інших.

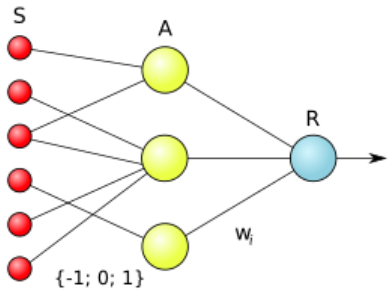


Рисунок 4 – Перцептрон

Послідовно передаючи дані від входів до виходу, КР напрацьовує рішення, ґрунтуючись на комбінації вхідних показників, параметрів сигналів, вироблених в середині мережі, та коефіцієнтах значимості сигналів.

Вихідний сигнал є рішенням, у якому наявні рекомендації зі встановлення пацієнту відсотків втрати працездатності, групи інвалідності, реабілітаційних заходів.

Запропонований підхід дасть змогу суттєво зменшити рутинну роботу з аналізу даних, пошуку рішення з огляду на комплексний стан особи. Також це допоможе мінімізувати вплив помилок експертів на прийняття рішення.

Список літературних джерел

1. Семенюк А. М., Ніколюк П. К. Моделювання систем прийняття рішень медичної експертизи. *Прикладні аспекти сучасних міждисциплінарних досліджень.* 2024. С. 45–47.

2. Rotshtein A. Design and tuning of fuzzy rule-based systems for medical diagnosis – Fuzzy and neuro-fuzzy systems in medicine, 2017.

3. Семенюк А. М. Створення реєстраційної системи МСЕК. *Матеріали LII науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2023):* (Вінниця, 21–23 червня 2023 р.). С. 744–746.

4. Перцептрон – джерела та інші електронні документи. *Вікіпедія.* URL: https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD (дата звернення: 22.11.2024).

УДК 004.415:004.774

Сидорчук Р. В., здобувач вищої освіти,
Потапова Н. А., канд. екон. наук, доцент,
доцент кафедри інформаційних технологій

ГЕКСАГОНАЛЬНА АРХІТЕКТУРА В РОЗРОБЦІ УНІВЕРСАЛЬНИХ ВЕБДОДАТКІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Гексагональна архітектура (Hexagonal Architecture), також відома як архітектура портів і адаптерів, є сучасним підходом до побудови програмних систем, що забезпечує високий рівень гнучкості та адаптивності. Основна ідея цієї архітектури полягає у чіткому розділенні бізнес-логіки системи від зовнішніх інтерфейсів, як-от користувацький інтерфейс, база даних, зовнішні сервіси чи API.

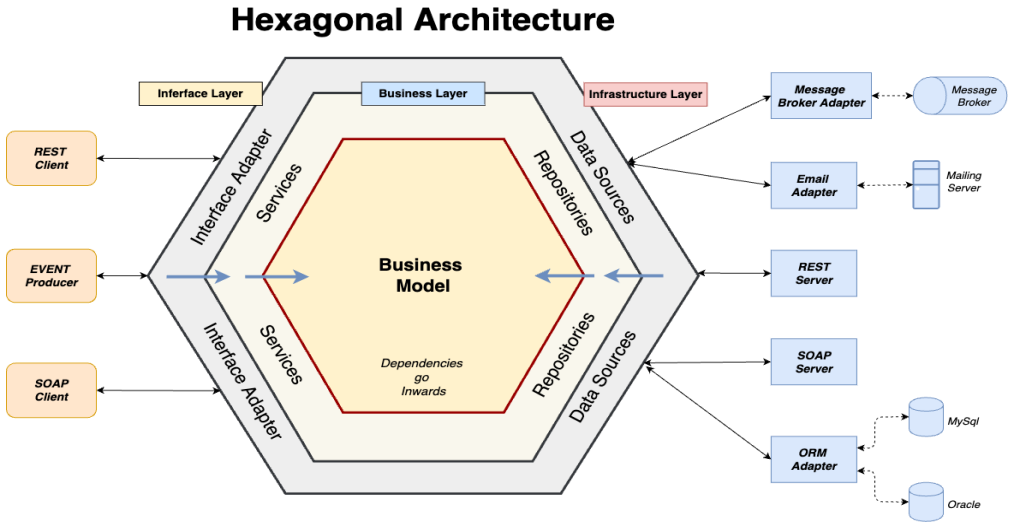


Рисунок 1 – Діаграма «Гексагональна архітектура»

Головною причиною вибору гексагональної архітектури для розробки універсального вебдодатка на основі системи підтримки прийняття рішень є її здатність ізолювати бізнес-логіку від змін у зовнішніх компонентах. Це критично важливо для систем, що передбачають аналіз і обробку великих обсягів даних, адже модулі можуть працювати незалежно від джерел даних чи специфічних інструментів, що використовуються для взаємодії з ними.

Переваги гексагональної архітектури:

- Модульність: кожен компонент системи може розроблятися, тестуватися та розширюватися незалежно.
- Легкість інтеграції: заміну бази даних чи зовнішнього API можна реалізувати без змін у бізнес-логіці.
- Тестованість: ізоляція компонентів спрощує валідацію бізнес-логіки.
- Масштабованість: зміни у зовнішніх компонентах не впливають на бізнес-логіку.
- Довгострокова підтримка: гексагональна архітектура сприяє впровадженню нових функцій без значних витрат.

Недоліки гексагональної архітектури:

- Складність: реалізація вимагає детального планування, що ускладнює старт проєкту.
- Ресурсоємність: потребує більше часу та ресурсів через створення додаткових компонентів.
- Складність для новачків: потребує навчання для глибшого розуміння принципів архітектури.

Гнучка структура гексагональної архітектури дає змогу легко адаптувати систему до нових умов, забезпечуючи високий рівень адаптивності. Вона є одним із найсучасніших підходів для побудови масштабованих вебдодатків, особливо в умовах динамічних вимог бізнесу.

Список використаних джерел

1. Ullman J. D., Widom J. A First Course in Database Systems. Pearson, 2015. 524 p.
2. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall, 2004. 624 p.
3. Литвиненко І. І. Технології програмування для сучасних вебплатформ. Київ: ІнфоПринт, 2021. 256 с.
4. Nardi B. A. A Study of Human Interaction with Technology. MIT Press, 2017. 300 p.
5. Гончарук О. Ю. Інформаційні системи підтримки прийняття рішень. Одеса: ОНУ, 2019. 272 с.
6. Microsoft. Building Cloud-Based Web Applications. Redmond: Microsoft Press, 2020. 384 p.

УДК 519.8:004.03

Яценко В. В., здобувач вищої освіти,
Сеник І. О., асистент кафедри
інформаційних технологій

ЗАСТОСУВАННЯ МЕТОДІВ ОПТИМІЗАЦІЇ ДЛЯ НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ ІЗ ВЕЛИКИМ ОБСЯГОМ ПАРАМЕТРІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Технології обробки даних критично важливі для галузей наукових досліджень та штучного інтелекту. Розвиток нейронних мереж, зокрема глибинного навчання, відкрив принципово нові можливості для розв’язання складних завдань розпізнавання патернів та прогнозування [1]. З кожною новою ітерацією розробки нейронні мережі переходили від простої до більш складної архітектури. Поступово штучний інтелект переходив від розгорнутої архітектури до згорткової, рекурентної, і наразі перебуває на етапі архітектури-трансформера [2]. Зі зміною архітектури експоненційно зростала і кількість їх параметрів. Сучасні моделі налічують сотні мільярдів параметрів, демонструючи стрімку тенденцію до ускладнення архітектур та збільшення обчислювальної потужності. Відтак створення сучасної нейронної мережі висуває особливі вимоги до процесів її навчання. Принципового значення набуває вибір стратегії оптимізації, яка безпосередньо впливає на швидкість збіжності, стабільність навчання та здатність мережі до узагальнення. Різні методи оптимізації демонструють різні підходи до цієї задачі, мають свої переваги та недоліки, які варто враховувати під час вибору методу для конкретного завдання. Фундаментальним елементом такого налаштування є функція втрат, яка кількісно оцінює розбіжність між передбаченнями моделі та реальними даними [3]. Правильно підібрана функція втрат дає змогу не лише оцінити точність моделі, але й забезпечити ефективне налаштування її параметрів під час навчання.

Метою дослідження стане аналіз алгоритмів оптимізації. Під час дослідження необхідно розглянути методи оптимізації, їх особливості, переваги та обмеження, які впливають на навчання великих нейронних мереж, надати ідею комбінованого методу і підсумувати відповідну інформацію.

У статті An optimization Strategy for Deep Neural Networks Training представлений стратегічний підхід до оптимізації навчання нейронних мереж із використанням сучасних адаптивних алгоритмів і методів регуляризації [4]. Автори акцентують на комбінованих алгоритмах, які поєднують адаптивне масштабування градієнтів і зменшення складності обчислень через ефективні стратегії розподілу параметрів. Результати досліджень показують, що ці інновації допомагають скоротити час навчання моделей і забезпечити стабільну збіжність.

Класичний градієнтний спуск є одним із найперших і базових підходів. У цьому методі оновлення параметрів моделі виконується на основі градієнта функції втрат щодо кожного параметра. Головна ідея полягає в тому, щоб рухатися в напрямку, протилежному градієнту, з метою зменшення значення функції втрат. Градієнтний спуск на основі всієї вибірки даних гарантує стабільність, але має значну обчислювальну складність, особливо для великих наборів даних. Це робить метод повільним і малопридатним для сучасних нейронних мереж.

Стохастичний градієнтний спуск (SGD) спрощує задачу, використовуючи випадковий зразок із набору даних на кожному етапі оновлення. Це значно прискорює процес навчання, але додає коливань у траєкторії збіжності, через що процес може бути менш стабільним. Міні-батч градієнтний спуск усуває цю проблему, даючи змогу використовувати невеликі групи даних для кожного кроку. Це знижує випадковість, характерну для SGD, і робить метод більш придатним для масштабних задач.

Адаптивні методи оптимізації значно розширили можливості навчання нейронних мереж. Один із найпопулярніших таких методів – Adam (Adaptive Moment Estimation). Він поєднує дві важливі ідеї: момент і адаптивне масштабування градієнтів. У цьому методі обчислюється експоненціальне згладжування середнього значення градієнтів і їх квадратів. Це допомагає Adam бути швидким і ефективним навіть для задач зі складними ландшафтами функції втрат. Проте Adam має тенденцію до нестійкості на пізніх етапах навчання, оскільки адаптивне масштабування може знижувати його здатність знаходити глобальний мінімум.

RMSProp – ще один адаптивний метод, який працює за принципом поділу градієнта на квадрат кореня середнього значення попередніх градієнтів [5]. Це дає змогу адаптивно змінювати швидкість навчання для кожного параметра, залежно від його важливості. RMSProp ефективний для задач із нерівномірно розподіленими градієнтами, але іноді може демонструвати повільнішу збіжність, порівняно з Adam.

Adagrad адаптує швидкість навчання для кожного параметра, зменшуючи її для тих параметрів, які часто оновлюються, і збільшуючи для рідко оновлюваних. Цей підхід добре працює для задач із розрідженими даними, наприклад, у роботі з текстом. Однак основний недолік Adagrad полягає у поступовому зменшенні швидкості навчання до дуже низьких значень, через що модель може перестати навчатися до досягнення мінімуму функції втрат.

Існують також модифікації градієнтного спуску, які додають до базового алгоритму додаткові компоненти для покращення процесу оптимізації. Момен-

тум є технікою, що використовує попередні значення градієнтів для визначення напрямку руху, додаючи інерцію до оновлення параметрів. Це допомагає швидше долати області з плоскими ландшафтами, прискорюючи збіжність. Nesterov Momentum покращує цей підхід, обчислюючи градієнт у точці, зрушеній на крок вперед за інерцією. Це дає змогу більш точно передбачати напрямок і знижує ризик надмірних коливань.

AdamW є вдосконаленням Adam, яке додає регуляризацію за допомогою зменшення ваг параметрів. Це дає змогу уникати перенавчання і зберігати стійкість процесу навчання навіть на великих наборах даних. Така модифікація робить AdamW одним із найкращих виборів для задач, пов'язаних із сучасними нейронними мережами.

Хоча кожен із цих методів має свої переваги, жоден із них не є ідеальним для всіх ситуацій. Об'єднання їхніх сильних сторін у новому підході може бути перспективним рішенням. Наприклад, метод, який комбінує адаптивність Adam, інерцію моментуму і регуляризацію AdamW, здатен забезпечити стабільність, швидкість та ефективність збіжності. Це особливо актуально для задач із великими обсягами даних і моделей із великою кількістю параметрів. Подібні комбіновані методи оптимізації можуть стати основою для навчання сучасних нейронних мереж із великими обсягами параметрів, даючи змогу покращити якість моделей і скоротити час їх навчання.

Отже, різноманітність методів оптимізації надає широкий спектр інструментів для навчання нейронних мереж. Кожен метод має свої переваги й обмеження, тому вибір залежить від специфіки задачі. Однак комбіновані підходи, які інтегрують сильні сторони кількох методів, відкривають нові можливості для ефективного навчання сучасних моделей. Впровадження таких алгоритмів може стати важливим кроком у розвитку оптимізації нейронних мереж, сприяючи підвищенню якості моделей і зменшенню витрат часу на їх навчання.

Список використаних джерел

1. What is Deep Learning? URL: https://cloud.google.com/discover/what-is-deep-learning (дата звернення: 28.11.2024).

2. What Is a Transformer Model? URL: https://blogs.nvidia.com/blog/what-is-a-transformer-model/ (дата звернення: 28.11.2024).

3. What's a loss function? URL: https://www.datarobot.com/blog/introduction-to-loss-functions/ (дата звернення: 28.11.2024).

4. Wu T., Zeng P., Song An C. optimization Strategy for Deep Neural Networks Training. URL: https://ieeexplore.ieee.org/document/10009665 (дата звернення: 28.11.2024).

5. Understanding RMSprop – faster neural network learning. URL: https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a (дата звернення: 28.11.2024).

СЕКЦІЯ 4
БАЗИ ДАНИХ І ЗНАНЬ. BIG DATA

УДК: 004.4:004.774]:001.895(043.2)

Ілик В. В., здобувач вищої освіти,
Якубич К. О., асистент кафедри
інформаційних технологій

ПРОГРЕСИВНІ ВЕБДОДАТКИ (PWA): ІННОВАЦІЙНІ МОЖЛИВОСТІ ДЛЯ ІНТЕРАКТИВНИХ ПЛАТФОРМ І ЇХ РОЛЬ У ТРАНСФОРМАЦІЇ КОРИСТУВАЦЬКОГО ДОСВІДУ

Анотація. Прогресивні вебдодатки (PWA) поєднують переваги вебдодатків і нативних додатків, забезпечуючи швидкість, надійність і кросплатформеність. Завдяки технологіям кешування, сервісним працівникам і офлайн-доступу PWA підвищують залученість користувачів і знижують витрати на розробку. Паралельно зростає підтримка PWA в браузерах, що відкриває нові можливості для бізнесу.

Ключові слова: прогресивні вебдодатки, PWA, кросплатформеність, кешування, сервісні працівники, офлайн-доступ, залученість користувачів.

Вступ. Прогресивні вебдодатки (PWA) стали основною технологією для розробки сучасних додатків завдяки своїй здатності працювати у вебсередовищі і надавати користувачам нативний досвід. Важливість PWA зросла у відповідь на зростаючі вимоги користувачів до швидкості завантаження та стабільності вебдодатків. У контексті цифрової трансформації, яка охоплює різні сектори економіки, PWA забезпечують новий підхід до створення додатків, який дасть змогу використовувати одне рішення для різних платформ і зберігати функціональність додатка навіть у разі перебоїв з інтернет-з'єднанням. Зокрема, PWA стають критично важливими в умовах обмеженого доступу до інфраструктури інтернету, особливо в регіонах, що розвиваються, де стабільний доступ до мобільного інтернету може бути проблематичним [1].

Прогресивні вебдодатки (PWA) поєднують у собі переваги як веб-, так і нативних мобільних додатків. Основними елементами PWA є маніфест, сервісні працівники і кешування. Маніфест визначає основні параметри програми, як-от іконка, назва та інші характеристики, що допомагають додатку бути встановленим на пристрій користувача як нативний. Сервісні працівники працюють у фоновому режимі, незалежно від вебсторінки, і обробляють мережеві запити, що дає змогу зберігати дані для офлайн-доступу. Використання інструментів кешування, як-от IndexedDB або Cache API, допомагає зберігати дані в браузері, що підвищує швидкість завантаження та зменшує споживання трафіка. Завдяки прогресивному покращенню PWA адаптуються до будь-якого пристрою чи браузера, використовуючи сучасні можливості, якщо вони доступні [2].

PWA є кросплатформенними за своєю природою, оскільки їх можна використовувати на будь-якому пристрої, який підтримує сучасні браузери. Це дає змогу компаніям уникнути необхідності розробляти окремі нативні додатки для різних операційних систем, як-от Android та iOS, що значно зменшує витрати на розробку та підтримку. Офлайн-режим забезпечується сервісними працівниками, які кешують дані та дозволяють користувачам продовжувати використовувати додаток навіть без активного інтернет-з'єднання. Це особливо

важливо для додатків, що потребують постійної доступності, наприклад, у сферах електронної комерції, логістики або медіа. Користувачі отримують миттєві оновлення контенту та інтерфейсу, що робить досвід роботи безперебійним і приємним. До того ж PWA мають значно менший розмір, порівняно з нативними додатками, оскільки вони не потребують установки через магазини додатків [3].

Покращення продуктивності та користувацького досвіду (UX/UI). Прогресивні вебдодатки (PWA) значно покращують користувацький досвід завдяки швидкому часу завантаження, миттєвим відповідям на взаємодію користувача та плавності роботи навіть за нестабільних умов мережі. Основною перевагою PWA є здатність кешувати дані та працювати в офлайн-режимі. Це означає, що користувачі можуть продовжувати взаємодіяти з додатком, навіть якщо з'єднання з інтернетом тимчасово відсутнє. Це дає користувачам відчуття стабільності й надійності додатка.

Додатковою інновацією є використання технології *lazy loading* (ліниве завантаження), яка завантажує тільки ту частину вебсторінки, яка наразі потрібна користувачу, знижуючи в такий спосіб час очікування та зменшуючи споживання інтернет-трафіка. PWA також підтримують push-сповіщення, які, попри відсутність активного використання додатка, дають змогу комунікувати з користувачем. Це допомагає підтримувати зв'язок із клієнтами і створювати персоналізований досвід. Наприклад, в електронній комерції push-сповіщення можуть повідомляти користувачів про знижки або зміни в замовленнях, підвищуючи конверсії та залученість.

Ще одним фактором, що підвищує рівень UX, є адаптивний дизайн. PWA автоматично підлаштовуються під різні розміри екранів, забезпечуючи однакову якість взаємодії як на смартфонах, так і на настільних комп'ютерах. Це означає, що користувач не відчуває розбіжностей у функціональності або інтерфейсі, коли переходить з одного пристрою на інший [4].

Економія коштів на розробку та обслуговування. PWA допомагають бізнесу значно скоротити витрати, пов'язані з розробкою та обслуговуванням. Замість того, щоб розробляти окремі нативні додатки для Android, iOS та інших платформ, бізнес може створити один PWA, який буде працювати однаково добре на всіх пристроях. Це знижує витрати на створення, тестування та підтримку додатка, оскільки не потрібно розробляти та оновлювати кілька версій для різних операційних систем. Оновлення PWA можуть бути виконані швидко і легко, оскільки вони поширюються через браузер, а не через магазини додатків, де часто потрібен час на перевірку та схвалення нових версій.

Збільшення залученості користувачів. PWA активно сприяють збільшенню рівня залученості користувачів через інтеграцію push-сповіщень, можливість роботи в офлайн-режимі та зручну навігацію. Push-сповіщення дають змогу бізнесу підтримувати постійний зв'язок із клієнтами, інформуючи їх про нові пропозиції, важливі оновлення чи події. Це підвищує конверсії, особливо в електронній комерції, де своєчасні сповіщення можуть стимулювати користувача повернутися до кошика покупок або скористатися знижкою.

Навіть більше, PWA значно менші за розміром, ніж нативні додатки, що означає, що користувачі не потребують багато місця на пристроях для їх уста-

новки. Це усуває бар'єр для встановлення, який часто зустрічається під час використанні нативних додатків. Така легкість доступу до програми та його установки прямо з браузера допомагає залучити більше користувачів і підвищити частоту взаємодії [5].

Технічні обмеження та перешкоди для впровадження. Однією з основних перешкод у впровадженні PWA є обмежена підтримка функціоналу на деяких операційних системах, особливо на iOS. Наприклад, на iOS сервісні працівники (service workers), які відповідають за роботу офлайн, мають обмежену підтримку. Також на платформі Apple push-сповіщення для PWA поки що не доступні, що зменшує функціональність, порівняно з нативними додатками.

Ще однією технічною перешкодою є доступ до апаратних функцій пристроїв, як-от камери, датчики або Bluetooth, що також може бути обмеженим або недоступним для PWA на різних платформах. Це обмежує використання PWA у сферах, де потрібен доступ до складних апаратних можливостей (наприклад, у додатках доповненої реальності або додатках, які використовують GPS та інші сенсори). Для таких застосувань нативні додатки все ще залишаються більш функціональними.

Перспективи розвитку PWA. Попри ці виклики, перспектива розвитку PWA виглядає доволі позитивно. З кожним новим релізом браузерів їх підтримка PWA розширюється, і з часом можна очікувати покращення доступу до апаратних можливостей та інших функцій. Навіть більше, великі технологічні компанії, як-от Google і Microsoft, продовжують активно підтримувати і просувати PWA, що підвищує ймовірність розширення їх можливостей та впровадження нових стандартів.

Ще одна перспектива розвитку PWA – це інтеграція зі штучним інтелектом (AI) для надання більш персоналізованого досвіду. AI може допомогти в автоматичному адаптуванні додатка під конкретного користувача, прогнозуванні поведінки користувачів, а також у вдосконаленні взаємодії через більш точні рекомендації та push-сповіщення.

Прогресивні вебдодатки (PWA) відкривають нові можливості для бізнесу та користувачів, поєднуючи найкращі аспекти традиційних вебсайтів і нативних мобільних додатків. Вони не тільки спрощують процес розробки та знижують витрати на підтримку, але й забезпечують високий рівень продуктивності та інтерактивності. Завдяки технологіям, як-от кешування та сервісні працівники, PWA працюють швидко і надійно навіть в умовах нестабільного інтернет-з'єднання, що робить їх незамінними для користувачів у різних умовах.

Список використаних джерел

1. Hume D. A. Progressive Web Apps: Building Fast, Reliable, and Engaging User Experiences. 2017. 200 p.

2. Chris Love. Progressive Web App Development by Example. 2018. P. 30–35.

3. Dean Hume. Progressive Web Apps: Building Fast, Reliable, and Engaging User Experiences. 2017. P. 40–45.

4. Tal Ater. Building Progressive Web Apps. 2017. P. 70–80.

5. Dean Hume. Progressive Web Apps: Building Fast, Reliable, and Engaging User Experiences. 2017. P. 85–95.

УДК 004.6

Рудь О. С., здобувачка вищої освіти,
Зелінська О. В., канд. техн. наук, доцент,
доцент кафедри інформаційних технологій

BIG DATA У ТРАНСПОРТНИХ МЕРЕЖАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Великі дані (Big Data) є невід'ємною частиною сучасних транспортних і логістичних мереж, адже відкривають нові можливості для оптимізації процесів і зниження витрат. Завдяки здатності збирати, обробляти й аналізувати величезні обсяги інформації компанії можуть приймати більш обґрунтовані рішення щодо маршрутизації, управління трафіком і прогнозування попиту. У галузі транспорту це дає змогу не лише покращити ефективність перевезень, а й значно зменшити час доставки та витрати ресурсів. Прогнозування заторів, оптимізація маршрутів та управління рухом за допомогою аналітики великих даних стають основними інструментами для створення сучасних інтелектуальних транспортних систем.

Big Data у транспортній логістиці – це обсяг інформації, яка збирається, обробляється та аналізується для поліпшення ефективності роботи транспортних і логістичних мереж. Вони включають різноманітні типи даних, як-от інформація з GPS-трекерів, сенсорів, систем моніторингу трафіка, мобільних додатків, а також дані соціальних мереж. Великий обсяг даних дає змогу створювати точніші моделі прогнозування, а також отримувати важливу інформацію для оперативного управління. Важливими характеристиками великих даних є їх обсяг, швидкість обробки і різноманітність, що забезпечує комплексний підхід до аналізу ситуацій у реальному часі [1].

Аналітика великих даних має важливу роль у зниженні логістичних витрат, даючи змогу компаніям ефективно керувати своїми ресурсами. Один із головних аспектів – це оптимізація маршрутів перевезень, яка дає змогу зменшити витрати на пальне та час доставки. Завдяки збору даних із GPS-систем, сенсорів і мобільних додатків можна точно прогнозувати найефективніші маршрути з урахуванням дорожньої ситуації, погодних умов та інших факторів. Наприклад, системи моніторингу трафіка в реальному часі можуть визначати найшвидший шлях, що допоможе знизити час у дорозі та витрати на пальне. До того ж Big Data допомагає у прогнозуванні попиту на транспортні послуги, що дає змогу зменшити перевантаження і забезпечити рівномірний розподіл вантажопотоків. Це також сприяє покращенню використання транспортних засобів і зниженню витрат на їх обслуговування. В результаті компанії можуть значно знизити операційні витрати, збільшити прибутковість і підвищити загальну ефективність своїх логістичних процесів.

Прогнозування заторів – ще один важливий аспект для ефективного управління транспортними мережами, і великі дані безпосередньо беруть у цьому участь. Використовуючи історичні дані про трафік, погодні умови, події на

дорогах та інші фактори, аналітика допомагає передбачити час та місця виникнення заторів. Збір даних у реальному часі з датчиків, GPS-трекерів і мобільних додатків дає змогу створювати точні моделі, що дають змогу водіям і транспортним компаніям своєчасно коригувати маршрути. До того ж використання алгоритмів машинного навчання допомагає врахувати варіативність трафіка та прогнозувати затори навіть за складних умов. Інтеграція таких даних у системи управління рухом допомагає зменшити затримки і покращити планування вантажоперевезень [2].

Використання великих даних для прогнозування і попередження аварій є ще однією сферою застосування аналітики в транспортних мережах. Аналізуючи історичні дані про дорожньо-транспортні пригоди, можна виявляти закономірності і фактори, що сприяють аварійним ситуаціям, як-от погодні умови, час доби, стан дороги чи інтенсивність руху. Збираючи в реальному часі дані з камер спостереження, датчиків руху та автомобілів, можна передбачити потенційно небезпечні ситуації та надіслати водіям попередження про можливі загрози. Інтелектуальні системи, що використовують алгоритми машинного навчання, здатні на основі цих даних оптимізувати швидкість руху та управлінські рішення, знижуючи ймовірність аварій. Навіть більше, аналітика великих даних допомагає органам влади визначати ділянки з високим рівнем аварійності та вжити заходів для їх покращення. Це включає встановлення додаткових знаків, зміну дорожніх умов або адаптацію світлофорних систем для зменшення ризику нещасних випадків.

Використання великих даних у транспортних мережах значно покращує ефективність логістичних процесів, оптимізуючи витрати та час доставки. Аналітика даних дає змогу прогнозувати затори, управляти трафіком і попереджати аварії, підвищуючи безпеку на дорогах. Завдяки інноваціям у сфері Big Data транспортні системи стають більш ефективними, зручними та безпечними для користувачів [3].

Список використаних джерел

1. Цифрові технології у логістиці. URL: https://blog.youcontrol.market/tsifrovitiekhnologhiyi-u-loghistitsi/ (дата звернення: 30.11.2024).

2. Журавльов О. В., Гармаш А. О. Удосконалення ІТ-забезпечення логістичної діяльності підприємств. 2020. С. 1–2.

3. Потапова Н. А., Волонтир Л. О., Зелінська О. В. Математичне та комп’ютерне моделювання функціонування логістичних процесів та систем. *Вісник Хмельницького національного університету. Технічні науки.* 2022. № 2. С. 73–80. URL: http://journals.khnu.km.ua/vestnik/?cat=65 (дата звернення: 30.11.2024).

СЕКЦІЯ 5
ПРИКЛАДНІ АСПЕКТИ ОБРОБКИ ДАНИХ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

УДК 004.9

Антонюк А. О., здобувачка вищої освіти,
Луценко А. В., д-р філософії з математики,
старший викладач, в. о. завідувача кафедри
прикладної математики та кібербезпеки

ЗАСТОСУВАННЯ МАТЕМАТИЧНОЇ СТАТИСТИКИ В АНАЛІЗІ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Анотація. У роботі йдеться про застосування математичної статистики в аналізі даних. Звернено увагу на її значущість у сучасному світі. Описано значення описової статистики, регресійного та кластерного аналізу, перевірки гіпотез і аналізу числових рядів. Та на основі цього підкреслено важливість математичної статистики для аналізу даних у сучасному світі.

Ключові слова: математична статистика, аналіз даних, регресійний аналіз, перевірка гіпотез, кластерний аналіз, аналіз числових рядів.

Вступ. Через стрімкий розвиток інформаційних технологій ми отримуємо доступ до великої кількості даних у різних сферах. Проте звичайне накопичення інформації не забезпечує її ефективність без застосування належних аналітичних методів. Дослідники використовують математичну статистику для отримання потрібної інформації з даних, прогнозування подій та оптимізації процесів.

Метою цієї роботи є вивчення методів математичної статистики, які використовуються для аналізу даних, і показ їх практичного значення в різних сферах діяльності.

Основний текст. Описова статистика. Основною метою описової статистики є опис набору даних. Вона використовує числові та графічні методи для пошуку закономірностей у наборі даних, узагальненні інформації і представленні її в зручній формі. Описовий статистичний аналіз використовує спеціальні інструменти для опису даних, до них належать частота, відсотки та показники центральної тенденції. Описова статистика дає змогу виявити загальні тенденції, перевірити гіпотези про розподіл даних та порівняти різні групи даних [1].

Регресійний аналіз. Регресійний аналіз – це статистичний метод, який використовується для моделювання зв’язку між залежною змінною та однією чи кількома незалежними змінними. Він широко застосовується в аналізі даних для визначення закономірностей, прогнозування та розуміння зв’язків між змінними. В аналізі даних регресійний аналіз використовується для визначення факторів, які впливають на зміни відповіді, і прогнозування майбутніх результатів. У фінансовому аналізі регресія використовується для оцінки впливу ринкових факторів на дохід компанії, а в медицині – для визначення впливу різних факторів на стан здоров’я людей [2].

Перевірка гіпотез. Перевірка гіпотез – це фундаментальна концепція математичної статистики, яка передбачає перевірку гіпотези про сукупність на основі вибірки даних. Вона широко використовується в аналізі даних, щоб

приймати обґрунтовані рішення та робити висновки щодо населення. Під час аналізу даних перевірка гіпотез використовується, щоб визначити, чи є результати випадковими, чи вони відображають реальний ефект [3].

Кластерний аналіз. Кластерний аналіз – це статистичний метод, який використовується для групування схожих об'єктів або випадків у кластери на основі їх характеристик. Він широко використовується в аналізі даних для виявлення шаблонів і структур у даних. Під час аналізу даних кластерний аналіз використовується для визначення базових сегментів у даних і прогнозування поведінки нових випадків. До того ж кластерний аналіз використовується для сегментації клієнтів у маркетингу чи виявлення шахрайства у фінансовій сфері [4].

Аналіз числових рядів. Аналіз числових рядів – це статистичний метод, який використовується для аналізу та прогнозування даних, зібраних за певний час. Математична статистика використовується для визначення закономірностей, тенденцій і аномалій у даних числових рядів, що дає змогу аналітикам робити прогнози та приймати обґрунтовані рішення. Аналіз числових рядів є важливим в аналізі даних, зокрема у фінансах та економіці допомагає досліджувати динаміку цін та акцій і прогнозувати економічні показники [5].

Висновки. Математична статистика відіграє ключову роль в аналізі даних і широко застосовується в різних галузях. Завдяки її методам, як-от описова статистика, регресійний та кластерний аналіз, перевірка гіпотез, існує можливість ефективно працювати з великими обсягами даних, виявляти закономірності та прогнозувати події.

Сучасні технології підвищили важливість статистичних методів, зробивши їх доступними для використання у різних сферах. Використання математичної статистики допомагає приймати обґрунтовані рішення, оптимізувати процеси та знаходити нові методи для вирішення складних задач.

Отже, застосування математичної статистики в аналізі даних є невід'ємною частиною сучасного інформаційного суспільства, що сприяє розвитку багатьох сфер людської діяльності.

Список використаних джерел

1. Applied A. Z. Statistics: Basic Principles and Application. *International Journal of Innovation and Economic Development.* 2021. № 7(3). P. 27–33.

2. Shahab Araghinejad Regression-Based Models Dordrecht: Springer, 2014. P. 49–83.

3. Sheldon M. Ross Chapter 8 – Hypothesis Testing, 2014. P. 297–356.

4. Voges K. E. Cluster Analysis Using Rough Clustering and k-Means Clustering. IGI Global, 2005. P. 435–438.

5. Velicer W. F., Fava J. L. Time Series Analysis. John Wiley & Sons, 2003. P. 581–606.

УДК 338.27:004.89

Баліцький В. В., здобувач вищої освіти,
Римар П. В., старший викладач кафедри
інформаційних технологій

ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ ПРОЦЕСАМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Актуальність теми дослідження визначається необхідністю модернізації управлінських процесів у ветеринарних клініках через зростання попиту на високоякісні послуги. Використання інформаційних систем дає змогу автоматизувати адміністративні, медичні та фінансові процеси, зменшити помилки та підвищити загальну продуктивність роботи персоналу.

Метою роботи є створення інформаційної системи, яка забезпечить ефективне управління процесами у ветеринарній клініці, включно з реєстрацією клієнтів, розкладом прийомів, веденням електронних медичних карток, управлінням запасами медикаментів та фінансовим обліком.

Основний текст. Управління ветеринарною клінікою включає широкий спектр завдань: від ведення записів пацієнтів до контролю за виконанням медичних протоколів. Традиційні способи управління є трудомісткими, піддаються впливу людського фактора і часто не відповідають сучасним вимогам точності та швидкості обробки даних. Інформаційні системи усувають ці недоліки шляхом автоматизації ключових процесів [1].

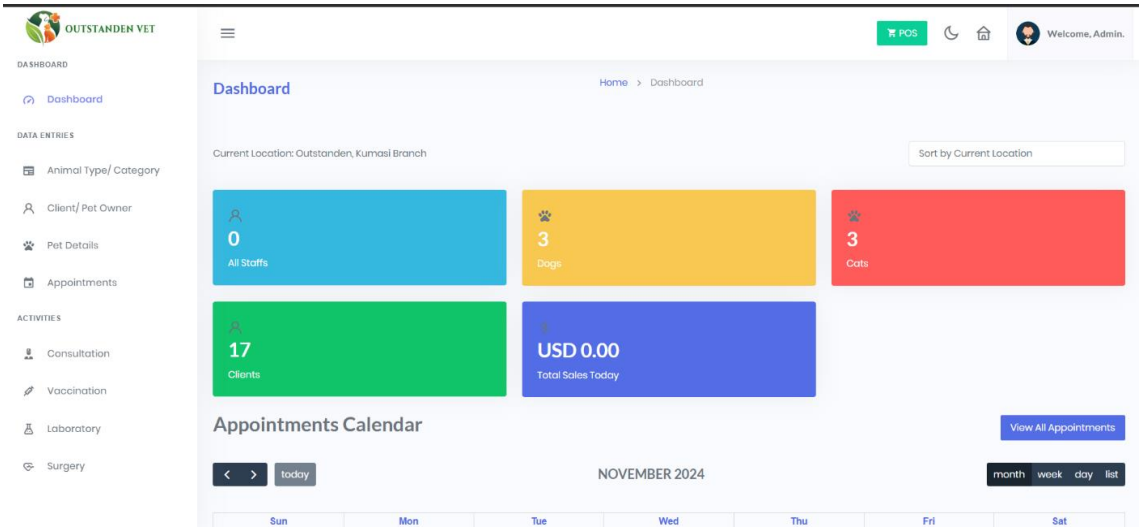


Рисунок 1 – Головна сторінка системи

Однією з головних функцій інформаційної системи є управління базою даних пацієнтів. Вона забезпечує збереження всієї необхідної інформації: анамнезу, результатів обстежень, призначень і процедур. Використання такої системи допомагає зменшити кількість паперової документації та уникнути втрати інформації [3].

Важливим аспектом є інтеграція систем управління запасами медикаментів. Ветеринарні клініки часто стикаються з проблемою недостатнього або надмірного запасу медикаментів, що впливає на якість обслуговування пацієнтів. Система автоматизує цей процес, попереджаючи персонал про необхідність поповнення запасів [2].

Ще одним критично важливим компонентом є модуль фінансового обліку. Він дає змогу слідкувати за витратами, формувати звіти та аналізувати фінансову діяльність клініки.

Для забезпечення гнучкості та зручності доступу сучасні інформаційні системи часто реалізуються як вебдодатки або хмарні рішення. Це допомагає користувачам працювати з даними незалежно від їх місцезнаходження та пристрою [4].

Висновки. Впровадження інформаційних систем у ветеринарні клініки сприяє покращенню якості послуг, оптимізації витрат і підвищенню продуктивності роботи персоналу. Завдяки автоматизації ключових процесів клініки можуть зосередитись на наданні високоякісної медичної допомоги, що позитивно впливає на лояльність клієнтів і конкурентоспроможність.

Список використаних джерел

1. Chapman P. Veterinary Practice Management: The Role of Information Systems. Springer, 2020.

2. Nguyen M. Veterinary Informatics and Automation Systems: Current Trends and Future Perspectives. Elsevier, 2019.

3. Чумак В. В., Стороженко М. В. Інформаційні системи в медицині. Київ: Наук. думка, 2021.

4. Сидоренко А. П. Автоматизація процесів управління клініками. Харків: Харківський національний університет, 2022.

УДК 005.8:005.94

Зелінський О. О., здобувач вищої освіти,
Потапова Н. А., канд. екон. наук, доцент
кафедри інформаційних технологій

ОСОБЛИВОСТІ ВИКОРИСТАННЯ ГІБРИДНИХ МЕТОДОЛОГІЙ УПРАВЛІННЯ ПРОЄКТАМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

В умовах швидких змін у бізнес-середовищі і зростаючих вимог до проєктів, що стосуються якості, строків і бюджету, традиційні методології управління проєктами часто виявляються недостатніми для досягнення оптимальних результатів. У відповідь на ці виклики виникли гібридні методології, які поєднують найкращі практики традиційних і гнучких підходів. Гібридний підхід дає можливість ефективно комбінувати стабільність і передбачуваність класичних методів з адаптивністю і гнучкістю Agile-методів, забезпечуючи високі результати у проєктному управлінні.

Гібридні методології управління проєктами об'єднують елементи традиційних підходів, як-от Waterfall, з практиками гнучких методів управління проєктами, наприклад, Scrum чи Kanban. Ці методології застосовуються залежно від вимог проєкту та особливостей його виконання, даючи змогу керівникам проєктів вибирати найбільш ефективні стратегії для кожного етапу.

- Waterfall (Каскадна модель) – це традиційний підхід до управління проєктами, де етапи виконання проєкту йдуть один за одним і чітко визначені.
- Agile – це набір принципів, що підкреслюють гнучкість, співпрацю та постійну адаптацію до змін, які можуть виникнути під час проєкту.

Гібридний підхід допомагає організаціям вибирати, які етапи проєкту будуть керуватися за допомогою класичних методів, а які – за допомогою гнучких, що дає можливість знижувати ризики та покращувати адаптивність до змін. Переваги гібридних методологій полягають у такому:

1. Баланс між плануванням і гнучкістю: гібридні методології дають змогу поєднувати чітке планування та розподіл ресурсів на ранніх етапах проєкту з гнучким підходом, що дає можливість адаптуватися до змін під час виконання. Це допомагає скоригувати план і стратегію в разі виникнення непередбачуваних змін або нових вимог.

2. Покращення управління ризиками: використання гібридних методологій дає змогу заздалегідь оцінити потенційні ризики, які можуть виникнути під час проєкту, а також створити стратегії для їх мінімізації. Гнучкість Agile – підходу дає можливість швидко коригувати проєкт у разі зміни обставин, що суттєво знижує ризики.

3. Збільшення залученості команди: гнучкі методи управління проєктами сприяють залученню всіх учасників процесу до ухвалення рішень і постійного обміну інформацією, що покращує взаєморозуміння і ефективність роботи команди. У гібридних моделях цей підхід поєднується з більш традиційними методами для контролю за результатами.

4. Підвищення якості продукту: оскільки гібридні методи допомагають на етапах виконання проєкту швидко коригувати дефекти або вдосконалювати продукт, це сприяє покращенню його якості. Зокрема, гнучкість Agile дає змогу постійно тестувати і покращувати кінцевий результат.

Для того, щоб правильно реалізувати гібридні методології, необхідно враховувати специфіку кожного проєкту, тип діяльності, а також потреби та можливості організації. Не існує універсального підходу, що б підходив для всіх ситуацій, тому важливо підібрати найбільш ефективну модель, що поєднує традиційні методи управління і гнучкі практики залежно від конкретних вимог і обставин.

Проєкт з розробки програмного забезпечення може бути прикладом використання гібридного підходу. На початкових етапах проєкту зазвичай потрібно визначити чіткі вимоги та бізнес-цілі, що потребують строгого планування. Це допомагає застосовувати методи класичного управління проєктами, які визначають основні етапи та строки. Однак у процесі розробки програмного продукту змінюються вимоги користувачів, з'являються нові технології або функціональні можливості, що вимагають адаптації. На цьому етапі доцільно використовувати методи Scrum або Kanban, які дають змогу швидко реагувати на зміни і адаптувати продукт до нових вимог.

Важливим складником такого підходу є інтеграція планування (Waterfall) та гнучкого розвитку продукту (Agile). Це допомагає досягти оптимальних результатів: чіткого визначення кінцевої мети і термінів, а також можливість врахування змін під час виконання проєкту. Завдяки використанню таких підходів можна мінімізувати ризики, пов'язані з відставанням від графіка або зміною вимог.

У випадку великих інфраструктурних проєктів, наприклад, будівництва великих об'єктів (мости, заводи, аеропорти) підхід Waterfall є більш доцільним на етапах проєктування, планування та будівництва, оскільки ці етапи мають чітко визначені вимоги і обмеження. Але на етапі тестування та здачі об'єкта в експлуатацію може бути доцільним використання гнучких методів для забезпечення високої якості і оперативного реагування на зміни чи дефекти, які можуть з'явитися під час тестування.

Зокрема, в таких проєктах методи Agile можуть бути використані для реалізації етапів проєкту, пов'язаних із дослідженнями та розробкою нових технологій або інновацій. Це дає змогу ефективно реалізувати інноваційні рішення, враховуючи нестабільність на етапах розробки і потребу швидкої адаптації до нових умов.

Запуск нового продукту на ринок може вимагати використання гібридного підходу, що поєднує досвід класичних методів планування і гнучкість для швидкої реакції на зміни. На початку проєкту важливо детально спланувати маркетингову стратегію, визначити цільову аудиторію та основні етапи розробки. Однак вже на етапі створення прототипу і тестування продукту в реальних умовах на ринку можуть виникнути нові, невизначені чинники, що вимагають швидкої реакції.

З цієї причини використання Scrum для тестування продукту або швидкого випуску мінімальних версій продукту є критично важливим. Водночас загалом підхід до управління проєктом буде залишатися класичним, оскільки важливо дотримуватися бізнес-цілей, бюджетів і термінів.

Порівняно з іншими, гібридні методології мають низку переваг:

1. Традиційні методи (Waterfall). Цей підхід є дуже ефективним для проєктів з чітко визначеними вимогами, де важливі суворі терміни і бюджети. Однак він має суттєві обмеження під час роботи з проєктами, де можуть виникати зміни. Перевага використання Waterfall полягає в чіткій структурі та послідовності етапів.

2. Гнучкі методи (Agile, Scrum). Вони забезпечують високу гнучкість, даючи змогу оперативно реагувати на зміни і коригувати перебіг проєкту під нові обставини. Але ці методи можуть бути менш ефективними для проєктів, де важливий жорсткий контроль і планування.

3. Гібридні методології. Порівняно з іншими підходами, гібридні методології надають максимальний баланс між гнучкістю та структурованістю. Вони допомагають зберегти чіткість і контроль на основних етапах проєкту і забезпечити гнучкість на етапах, де це найбільш необхідно.

Вибір методології також залежить від корпоративної культури компанії. У компаніях, де велике значення має планування і контроль, традиційні підходи

можуть бути більш ефективними, тоді як в організаціях із гнучкою та адаптивною культурою гібридні методології можуть призвести до більш ефективного результату. Інтеграція традиційних і гнучких методів вимагає відповідної підготовки керівників проєктів і членів команди, оскільки це потребує високого рівня координації та розуміння того, коли і як застосовувати той чи інший підхід. Тому організації, які вирішують використовувати гібридні методології, повинні інвестувати в навчання та розвиток своїх працівників, а також у вдосконалення комунікаційних процесів.

Отже, гібридні методології управління проєктами є ефективними інструментами для організацій, які працюють в умовах високої невизначеності та змін. Вони дають змогу об'єднувати чіткість і планування традиційних методів з гнучкістю і швидкою адаптацією гнучких підходів, що дає можливість більш ефективно реагувати на зміни під час виконання проєктів.

Цей підхід є особливо корисним у проєктних сферах, де важливо забезпечити як контроль за виконанням, так і швидкість реагування на нові вимоги або непередбачувані зміни. Однак для успішного впровадження гібридних методологій необхідно враховувати специфіку кожного проєкту, а також готовність організації до змін і адаптації під нові підходи в управлінні.

Список використаних джерел

1. Phillips J. IT Project Management: On Track from Start to Finish. Fourth Edition. McGraw Hill Professional, 2017. 557 p.

2. Schwalbe K. Information Technology Project Management. 9 ed. Cengage Learning, 2018. 672 p.

3. Harold K. Project Management: A Systems Approach to Planning, Scheduling, and Controlling 13th edition. John Wiley & Sons Inc, 2022. 880 p.

УДК 004.42

Ковальчук А. А., здобувач вищої освіти,
Римар П. В., старший викладач кафедри
інформаційних технологій

ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ MAPLE ДЛЯ ПОБУДОВИ КРИВИХ ТА ПОВЕРХОНЬ ТРЕТЬОГО ПОРЯДКУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Програмне забезпечення Maple є провідним інструментом для математичного моделювання, що пропонує великий функціонал для побудови графіків та візуалізації складних математичних об'єктів. Його застосування включає побудову кривих та поверхонь третього порядку, що знаходить застосування в інженерії, фізиці та інших галузях науки [1].

Актуальність. Побудова кривих та поверхонь третього порядку відіграє важливу роль у моделюванні фізичних систем, дослідженні геометричних властивостей об'єктів та розв'язанні прикладних задач. Використання Maple дає змогу швидко створювати тривимірні зображення, автоматизувати процес аналізу та

інтерактивно працювати з параметрами моделей. Це особливо важливо у задачах, де графічна інтерпретація є основою для прийняття рішень.

Мета дослідження. Мета цієї роботи полягає у дослідженні можливостей Maple для побудови та аналізу кривих і поверхонь третього порядку. Це включає як створення візуалізацій, так і аналіз їх геометричних властивостей для реальних практичних задач.

1. Побудова кривих третього порядку, їх аналіз та інтерпретація.

2. Створення тривимірних поверхонь третього порядку та оцінка їх властивостей.

3. Застосування побудованих моделей для розв'язання задач у прикладних областях, як-от технічне моделювання, програмування та природничі науки.

Основний матеріал та результати. Криві третього порядку описуються рівнянням:

$$y = ax^3 + bx^2 + cx + d,$$

де a, b, c, d – коефіцієнти, що визначають форму кривої [3].

У Maple функція *plot* дає змогу легко побудувати такі графіки. Наприклад, для кривої:

$$y = 2x^3 - 3x^2 + x - 1.$$

Графік зображено на рис. 1.

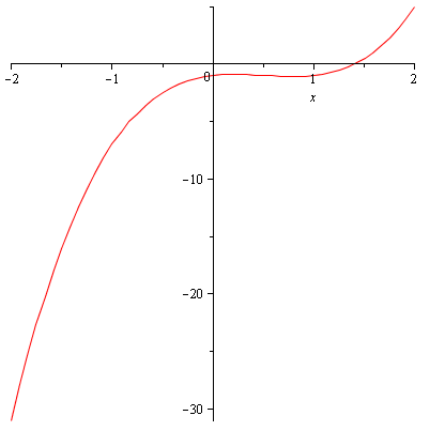


Рисунок 1 – Графік кривої

Лістинг 1

```
with(plots):
plot(2*x^3 - 3*x^2 + x - 1, x = -2 .. 2, color = red);
```

Аналіз показав, що крива має дві точки перегину, які можна знайти за допомогою команди *diff*. Отримані графіки є важливим інструментом для аналізу динамічних процесів у фізиці, наприклад, траєкторії частинок у полі сил.

Поверхні третього порядку задаються рівнянням:

$$z = ax^3 + by^3 + cxy + dx^2y + exy^2 + f,$$

де коефіцієнти a, b, c, d, e визначають форму поверхні.

Для прикладу, поверхня побудована у Maple за допомогою команди plot3d на рис. 2.

$$z = x^3 - y^3 + 3xy.$$

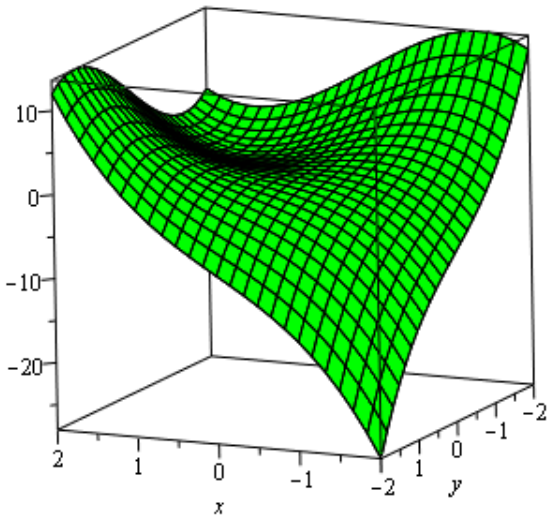


Рисунок 2 – Графік поверхні

Лістинг 2

```
with(plots):
plot3d(x^3 - y^3 + 3*x*y, x = -2 .. 2, y = -2 .. 2, axes = boxed, color = green);
```

Візуалізація показала цікаву топологію поверхні, зокрема наявність сідлових точок. Такий підхід може використовуватись у моделюванні теплових потоків чи силових полів.

Приклад використання. Для задачі моделювання поверхні хвильового фронту $z = 2x^3 + 3y^3 - xy^2$ у фізиці була побудована модель, яка дає змогу дослідити зміну амплітуди хвиль у різних точках простору (рис. 3). Аналіз моделі вказав на наявність симетрії, що суттєво спрощує подальший аналіз.

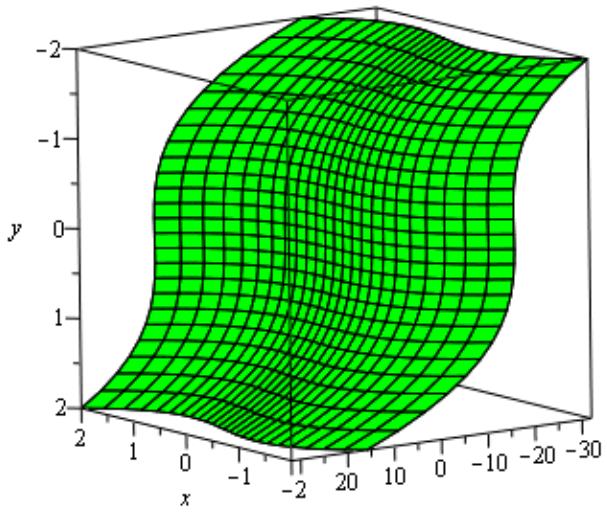


Рисунок 3 – Графік поверхні хвильового фронту

Висновки. Використання Maple для побудови кривих та поверхонь третього порядку підтвердило ефективність програмного забезпечення у вирішенні завдань цього типу. Інструменти Maple забезпечують високу точність, швидкість і наочність, що є особливо важливим для прикладних досліджень. Програмне забезпечення Maple рекомендоване для використання у науці, освіті та інженерії [2].

Список використаних джерел

1. Maplesoft. Visualization Tools in Maple. 2023. URL: https://www.maplesoft.com/products/maple/features/visualization.aspx (дата звернення: 22.11.2024)
2. Maplesoft. Examples from Maple Conference. 2023. URL: https://www.maplesoft.com/mapleconference/2023 (дата звернення: 22.11.2024)
3. Визначення кривої третього порядку. URL: https://bit.ly/49ec9K6 (дата звернення: 22.11.2024).

УДК 004.77:[005.8:338.28]:061.2

Костенко Р. О., здобувач вищої освіти,
Зелінська О. В., канд. техн. наук, доцент,
доцент кафедри інформаційних технологій

РОЛЬ ПРОЄКТНОГО МЕНЕДЖЕРА В ЦИФРОВІЙ ТРАНСФОРМАЦІЇ МІЖНАРОДНИХ ОРГАНІЗАЦІЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасний світ стикається з багатьма викликами, які створюють умови глобальної нестабільності. Економічні кризи, політичні конфлікти, пандемії та технологічний прогрес змінюють правила гри в бізнесі, науці, освіті та державному управлінні. У цьому контексті цифрова трансформація стає важливим інструментом для міжнародних організацій, що прагнуть зберегти конкурентоспроможність і ефективність. Проєктний менеджер відіграє ключову роль у забезпеченні успішного впровадження цифрових змін, координуючи ресурси, адаптуючи підходи до управління проєктами та нівелюючи вплив нестабільності на результати [1].

Цифрова трансформація охоплює широкий спектр змін, включно зі впровадженням хмарних технологій, автоматизацією бізнес-процесів, інтеграцією штучного інтелекту (AI) та великих даних (Big Data). Наприклад, міжнародні організації, як-от ООН чи ЮНЕСКО, активно використовують цифрові платформи для координації глобальних ініціатив, наприклад, боротьби зі зміною клімату чи просуванням освіти. У цьому контексті проєктні менеджери повинні враховувати не лише технічні аспекти, але й соціальні, етичні та культурні чинники впровадження цифрових змін. Глобальна нестабільність значно ускладнює процес цифрової трансформації. Постійні зміни умов потребують від проєктних менеджерів швидкого реагування на непередбачувані обставини. Економічна нестабільність обмежує доступ до ресурсів, змушуючи оптимізувати бюджети та пріоритети. До того ж політичні конфлікти та різноманітність куль-

тур ускладнюють координацію між міжнародними командами, особливо якщо проєкт охоплює бізнес, науку, освіту та владу. Усі ці фактори вимагають від проєктного менеджера не лише високих професійних навичок, але й здатності до адаптації та стратегічного мислення. Згідно зі звітом McKinsey & Compan, приблизно 70 % проєктів цифрової трансформації зазнають невдачі через неправильне планування, недостатню підтримку з боку керівництва або опір змінам. Наприклад, одна з найбільших міжнародних логістичних компаній провалила впровадження нової системи управління через недостатню підготовку співробітників до роботи з новим програмним забезпеченням. Такі приклади підкреслюють важливість проєктного управління змінами та врахування людського фактора [2].

У таких умовах ключовим інструментом для проєктних менеджерів стають гнучкі методології управління, зокрема Agile, Scrum і Kanban. Ці підходи забезпечують адаптивність до змін, дають змогу швидко переглядати пріоритети та впроваджувати нові рішення без затримок. Наприклад, використання Agile-підходу дає змогу розбити великий проєкт на короткі ітерації, що допомагає регулярно оцінювати прогрес і вчасно реагувати на зовнішні зміни. Методології Scrum і Kanban створюють прозору систему управління задачами, завдяки якій усі учасники проєкту мають доступ до актуальної інформації та чітко розуміють свої обов'язки. Проєктний менеджер також відповідає за формування та координацію мультидисциплінарних команд, які включають представників бізнесу, науки, освіти та влади. Це особливо важливо в міжнародних проєктах, де кожен сектор вносить свій унікальний вклад у досягнення спільної мети. Наприклад, освітні та наукові установи можуть надати знання і технології, бізнес – ресурси та комерційний досвід, а державні органи – підтримку на законодавчому рівні. Проєктний менеджер виступає посередником, створюючи умови для ефективної комунікації та побудови довіри між усіма сторонами. Яскравим прикладом інтеграції різних секторів є програма «Горизонт Європа» (Horizon Europe), яка фінансує науково-дослідні проєкти в ЄС. У межах цієї програми бізнес, університети та державні установи співпрацюють у галузі зеленої енергетики та цифровізації промисловості. Проєктні менеджери відіграють важливу роль у таких ініціативах, забезпечуючи координацію між партнерами, управління бюджетами та дотримання графіків [3].

Окрім методологій, проєктний менеджер активно використовує сучасні цифрові інструменти для управління проєктами. Платформи Jira, Trello та Microsoft Project допомагають автоматизувати процеси, слідкувати за виконанням задач і забезпечувати прозорість роботи команд. Для покращення комунікації та управління даними міжнародні організації все частіше використовують інтеграційні платформи, як-от Slack і Monday.com. Хмарні сервіси, як-от Google Workspace і Microsoft 365, забезпечують безпечне зберігання даних та спрощують віддалену роботу. Наприклад, у період пандемії COVID-19 багато університетів перейшли на дистанційну освіту, використовуючи Zoom і Moodle, що дало змогу зберегти навчальний процес. Аналітичні інструменти допомагають оцінювати ризики та прогнозувати вплив змін на проєкт. Наприклад, у проєктах, пов'язаних зі впровадженням дистанційної освіти, такі інструменти сприяють

ефективному плануванню та моніторингу виконання завдань, навіть якщо команди працюють у різних країнах [4].

Роль проєктного менеджера в умовах цифрової трансформації міжнародних організацій є вирішальною, особливо в умовах нестабільності. Саме проєктний менеджер забезпечує координацію, адаптацію та впровадження інноваційних рішень, що сприяють успішній інтеграції різних секторів. Використовуючи гнучкі методології, сучасні цифрові інструменти та стратегічний підхід до управління, він допомагає організаціям залишатися стійкими до зовнішніх викликів та досягати поставлених цілей. У майбутньому роль проєктного менеджера лише зростатиме, оскільки цифрова трансформація стає необхідною умовою для успішного функціонування в сучасному світі. У найближчі роки очікується ще більше зростання інвестицій у цифрову трансформацію. За прогнозами Gartner, витрати на цифрові технології до 2025 р. перевищать 10 трильйонів доларів. Успішна реалізація цих проєктів залежатиме від компетентності проєктних менеджерів, які мають забезпечити не лише технічне впровадження, а й культурну інтеграцію цифрових рішень у щоденну роботу організацій.

Список використаних джерел

1. Огляд інструментів цифрової трансформації економіки України. URL: https://shorturl.at/UyBnz (дата звернення: 25.11.2024).
2. The Role of Project Managers in Digital Transformation Initiatives. URL: https://shorturl.at/nlF5i (дата звернення: 25.11.2024).
3. Project Management in the Era of Digital Transformation. URL: https://shorturl.at/3r0Ws (дата звернення: 22.11.2024).
4. Концепція цифрової трансформації освіти і науки. URL: https://shorturl.at/ZB1xV (дата звернення: 22.11.2024).

УДК 519.85

Круцюк Д. О., здобувач вищої освіти,
Римар П. В., старший викладач кафедри
інформаційних технологій

ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ MAPLE ДЛЯ РОЗВ’ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Програмне забезпечення Maple є потужним інструментом для математичного моделювання, аналізу та розв’язання задач оптимізації. Воно забезпечує інтеграцію з алгебраїчними методами, чисельними алгоритмами і має розвинені функції візуалізації. Maple знаходить застосування в інженерії, економіці, природничих і точних науках, допомагаючи вирішувати складні прикладні завдання [1].

Актуальність. У сучасному світі постійно зростає потреба в оптимізації процесів, які стосуються управління ресурсами, зниження витрат, збільшення

продуктивності й підвищення ефективності різних систем. Програмне забезпечення Maple забезпечує високий рівень автоматизації процесів розрахунків, що не тільки знижує ймовірність помилок, але й дає змогу швидко адаптуватися до змін у зовнішніх умовах. Наприклад, у промисловості це може включати оптимізацію логістичних ланцюгів, витрат матеріалів або енергетичних ресурсів, тоді як у науці Maple використовується для моделювання складних фізичних і хімічних процесів [1, 2].

Аналіз останніх досліджень. Сучасні наукові публікації та презентації на конференціях, зокрема Maple Conference 2023, висвітлюють численні приклади використання Maple у задачах оптимізації. Особливу увагу приділено алгоритмам LPSolve (для задач лінійного програмування) і NLPSolve (для нелінійної оптимізації). До того ж у роботах останніх років наголошується на ефективності Maple у фінансовому аналізі, наприклад, у моделюванні динаміки фондового ринку та в біоінженерії для оптимізації технічних процесів. Завдяки постійним оновленням функціоналу Maple відповідає вимогам сучасної науки і техніки [1, 3].

Основною метою роботи є детальне вивчення можливостей Maple для вирішення задач оптимізації різних типів, включно з лінійними і нелінійними програмуванням. Це дослідження спрямоване на інтеграцію теоретичних і прикладних підходів із застосуванням алгоритмів Maple для вирішення реальних прикладних завдань [1].

Поставлені задачі включають:

1. Оптимізацію виробничих витрат за умов обмеженості ресурсів.

2. Побудову моделі нелінійної оптимізації технічної системи зі врахуванням складних залежностей параметрів.

Лінійна оптимізація формується так:

$$minZ = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

де c_i – коефіцієнти витрат, x_i – обсяги ресурсів.

Обмеження задаються у вигляді:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1.$$
$$x_i \geq 0.$$

Нелінійна оптимізація описується функцією:

$$minf(x);$$
$$g_i(x) \leq 0;$$
$$h_j = 0,$$

де $f(x)$ – нелінійна цільова функція, $g_i(x), h_j(x)$ – обмеження [1].

Лінійне програмування. Для задачі мінімізації витрат на транспортну логістику використано команду LPSolve у Maple. Наприклад, під час оптимізації перевезення товарів між трьома містами розв’язок має вигляд:

$$x_1 = 30;\ x_2 = 20;\ x_3 = 50;\ Z = 1\,500\ [2].$$

Візуалізація рішення у двовимірному просторі підтвердила оптимальність результату, що важливо для інтерпретації [3].

Нелінійне програмування. У задачі мінімізації енергоспоживання двигуна функція енерговитрат мала вигляд:

$$f(x_1, x_2) = x_1^2 + x_2^2 - x_1 x_2 + 3$$

за обмежень:

$$x_1 + 2x_2 \leq 10;$$
$$x_{1,2} \geq 0.$$

Maple дає змогу знайти глобальний мінімум функції:

$$x_1 = 2{,}5;\ x_2 = 3{,}75; \quad f(x) = 12{,}125\ [2].$$

Отриманий розв'язок візуалізовано за допомогою тривимірної діаграми.

Для задач лінійного програмування було проведено аналіз проблеми мінімізації транспортних витрат. Maple дає змогу швидко отримати точний результат за допомогою алгоритму LPSolve, що є частиною його оптимізаційного пакету. У прикладі, що розглядає розподіл товарів між трьома містами, отримані оптимальні значення змінних продемонстрували ефективність алгоритму. Додаткова візуалізація рішень підтвердила їх наочність і зручність для аналізу [2].

У задачах нелінійної оптимізації, як-от мінімізація енергоспоживання технічних систем, використання команди NLPSolve допомогло врахувати складні взаємозв'язки між параметрами. Наприклад, мінімізація функції витрат енергії за обмежених ресурсів дала змогу знайти глобальний мінімум, який додатково було підтверджено графічними методами у тривимірному просторі [3].

Головна ідея дослідження полягає у демонстрації ефективності та універсальності Maple для вирішення різноманітних типів оптимізаційних задач. Завдяки потужному інструментарію Maple дає змогу не тільки підвищувати точність розрахунків, а й автоматизувати процеси аналізу, забезпечуючи економію часу і ресурсів. Використання Maple у сферах, як-от фінанси, логістика, технічне моделювання, робить його незамінним інструментом у сучасній науці та індустрії [1, 2].

Дослідження підтвердило, що Maple є потужним інструментом для прикладних задач оптимізації, який сприяє скороченню витрат часу і ресурсів, а також підвищенню точності отриманих рішень.

Список використаних джерел

1. Maplesoft. Maple Optimization Tools. 2023. URL: https://www.maplesoft.com/products/maple/features/optimization.aspx (дата звернення: 22.11.2024).

2. MaplePrimes. Optimization Examples and Tutorials. 2023. URL: https://www.mapleprimes.com/DocumentFiles/238347_question/optimization_2023.pdf (дата звернення: 22.11.2024).

3. Maplesoft. Presentations from Maple Conference. 2023. URL: https://www.maplesoft.com/mapleconference/2023/ (дата звернення: 22.11.2024).

УДК 004.451.4:004.75:614.2

Курдупов О. Л., здобувач вищої освіти,
Антонов Ю. С., канд. фіз-мат. наук,
доцент кафедри інформаційних технологій

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕДИЧНОГО ЗАКЛАДУ НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Донецький національний університет імені Василя Стуса, м. Вінниця

Завдяки своєму потенціалу технологія блокчейну знайшла своє застосування у різних галузях, де важливість забезпечення конфіденційності та цілісності інформації, відкритості, достовірності та незмінності даних виявилася найвищою. Зокрема, його потенціал виявляється у медичній сфері, де він може зробити значний внесок у покращення ефективності та безпеки медичного обслуговування [1].

Принцип роботи інформаційної системи медичного закладу на основі технології блокчейн, спрямованої на захист від несанкціонованого втручання та конфіденційність даних користувачів, у проєкті побудований, містячи такі кроки:

- Децентралізоване зберігання даних. Дані пацієнтів (медичні записи, рецепти, діагнози) не зберігаються централізовано. Замість цього у блокчейні зберігаються хеші даних та посилання на зашифровані файли в зовнішніх сховищах [2]. Саме це унеможливлює несанкціоноване видалення або зміну даних, оскільки кожна транзакція фіксується у незмінній формі.
- Контроль доступу. Система використовує рольову модель доступу (RBAC), у якій пацієнти, лікарі та адміністратори мають чітко визначені права [4]. Доступ до медичних записів здійснюється через смарт-контракти, які визначають, хто і коли може переглядати або редагувати дані. Тому до чутливої інформації доступ можуть отримати тільки уповноважені особи.
- Прозорість та аудит. Кожна дія з медичними записами (створення, редагування, перегляд) фіксується у блокчейні. Журнал неможливо змінити або видалити, що забезпечує повну прозорість.
- Зашифровані дані. Дані пацієнтів зберігаються у зашифрованому вигляді. Для доступу використовується приватний ключ пацієнта, який роздається лише уповноваженим лікарям.
- Ідентифікація та автентифікація. Кожен користувач отримує унікальний цифровий сертифікат, виданий Hyperledger Fabric CA, який використовується для входу до системи [3]. Також використовується багатофакторна автентифікація (пароль + біометрія або SMS-код) для доступу до системи.
- Верифікація правдивості даних. Кожен медичний запис має унікальний хеш, який зберігається у блокчейні. Це дає змогу перевірити, чи були дані змінені або пошкоджені. Система порівнює хеш поточного запису з хешем у блокчейні для виявлення будь-яких змін.
- Резервне копіювання та відновлення. Дані дублюються між вузлами блокчейну, що забезпечує їх доступність навіть у разі технічних збоїв.

Захист даних у проєкті здійснюється за допомогою контрольної суми. Цей процес використовується для перевірки цілісності даних і виявлення будь-яких змін, включно з помилками під час передавання або зловмисні дії [5].

Коли дані створюються або додаються в систему, на їх основі обчислюється контрольна сума через алгоритм SHA-256. Контрольна сума зберігається окремо у базі даних. Під час доступу до даних або їх редагування контрольна сума обчислюється заново, і її значення порівнюється з оригінальною. Якщо контрольна сума збігається, дані вважаються цілісними. Детальна схема етапів роботи з контрольними сумами зображена на рис. 1.

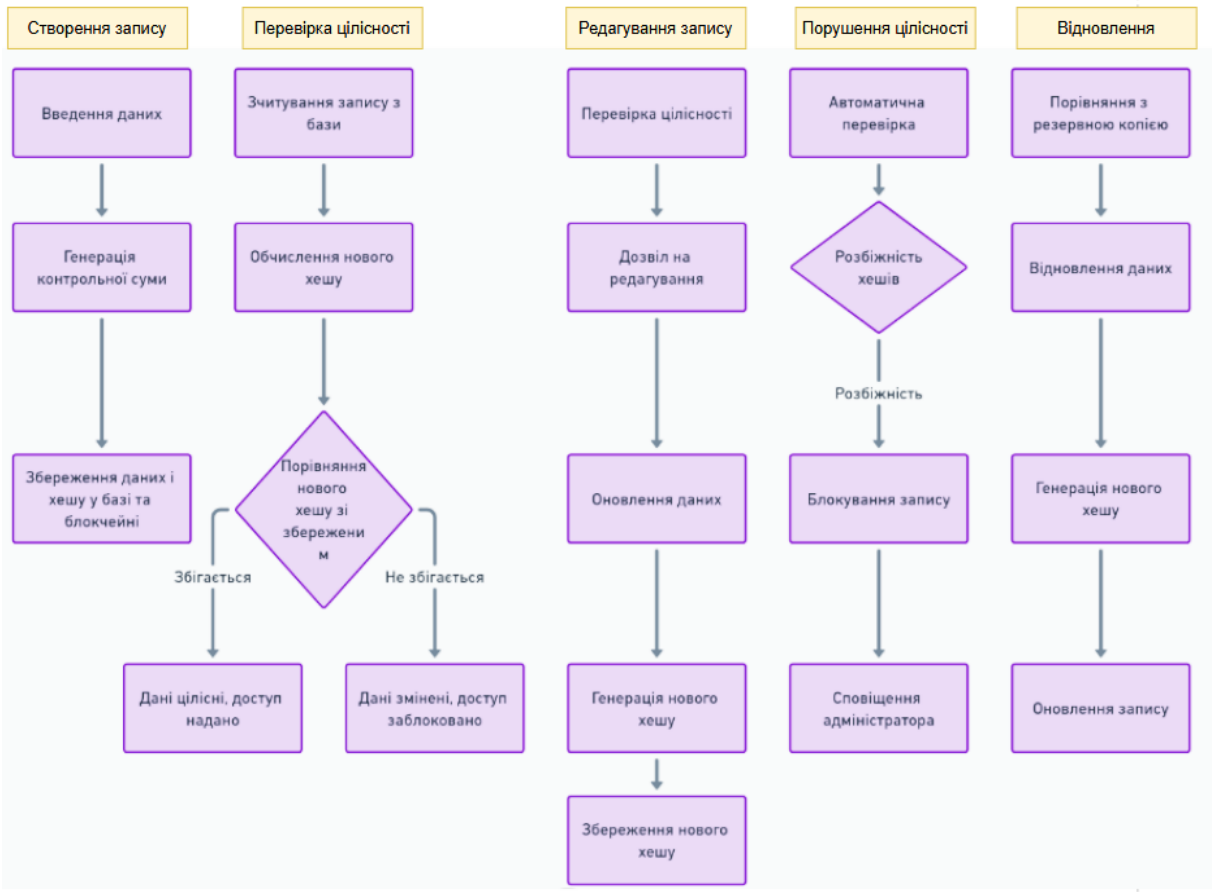


Рисунок 1 – Схема роботи з контрольними сумами

Якщо запис був змінений, нова контрольна сума обчислюється після редагування, а стара контрольна сума замінюється новою в базі даних.

Використання контрольних сум сприяє збереженню цілісності та безпеці системи через миттєве виявлення змін у даних. Також контрольна сума забезпечує захист від випадкових чи навмисних змін, усі операції логуються. Перевагами є і масштабованість (підходить для великих обсягів даних) і легкість реалізації (алгоритми хешування (SHA-256) доступні у стандартних бібліотеках). Завдяки автоматизації всі процеси перевірки працюють автоматично. Цей підхід гарантує, що дані залишатимуться незмінними, а всі спроби їхнього модифікування будуть швидко виявленні.

Список використаних джерел

1. Мацик О. В. Модернізація інформаційно-комунікаційного забезпечення діяльності медичних установ. *Інноваційна економіка*. 2020. № 5–6. С. 97–102. URL: http://inneco.org/index.php/innecoen/article/view/657/726 (дата звернення: 17.11.2024).
2. Автоматизована медична інформаційна система / В. О. Юхимець, В. Г. Терентюк, В. А. Науринський, В. В. Куц, В. В. Яровий, А. С. Єрьоміна, О. Л. Мельник, О. С. Лісневич. *Впровадження та оптимальні рішення.* 2016. URL: http://www.ifp.kiev.ua/ftp1/original/2016/yukhymets2016-8 (дата звернення: 20.11.2024).
3. Blockchain-Based Medical Records Secure Storage and Medical Service Framework / Y. Chen, S. Ding, Z. Xu, H. Zheng, S. Yang. *Journal of Medical Systems.* Vol. 43, № 1, Nov. 2018. DOI: 10.1007/s10916-018-1121-4 (дата звернення: 08.11.2024).
4. Sillaber C., Waltl B. Life Cycle of Smart Contracts in Blockchain Ecosystems. Datenschutz und Datensicherheit. 2017. Vol. 41. DOI: 10.1007/s11623-017-0819-7 (дата звернення: 08.11.2024).
5. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts / A. Kosba, A. Miller, E. Shi, Z. Wen. S. Papamanthou. *IEEE Symposium on Security and Privacy (SP).* San Jose, CA, USA. 2016. P. 839–858. DOI: 10.1109/SP.2016.55 (дата звертання: 10.11.2024).

УДК 004.77:[005.8:338.28]:061.2

Куцмай В. Я., здобувач вищої освіти,
Зелінська О. В., канд. техн. наук, доцент,
доцент кафедри інформаційних технологій

УПРАВЛІННЯ МУЛЬТИНАЦІОНАЛЬНИМИ КОМАНДАМИ В УМОВАХ ЦИФРОВОЇ ЕКОНОМІКИ ТА НЕСТАБІЛЬНОСТІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Цифрова економіка змінила підходи до організації праці, відкривши безпрецедентні можливості для роботи мультинаціональних команд. Завдяки цифровим технологіям географічні межі більше не обмежують компанії, а глобалізація дає змогу залучати талановитих спеціалістів з усього світу. Проте управління такими командами є складним завданням, особливо в умовах нестабільності, коли економічні, соціальні чи політичні фактори можуть суттєво впливати на продуктивність і згуртованість колективу. Проджект-менеджер, як ключова фігура в організації роботи, повинен враховувати ці виклики і використовувати ефективні інструменти та підходи для досягнення поставлених цілей.

Одним із головних викликів є культурні відмінності. Мультинаціональні команди об’єднують фахівців із різних країн, які мають свої унікальні підходи до спілкування, прийняття рішень і управління часом. У нестабільних умовах, коли напруження може бути вищим, проджект-менеджеру особливо важливо формувати відкриту та прозору комунікацію, яка враховує культурні особливості кожного члена команди. Для цього необхідно впроваджувати регулярні зустрічі, проводити навчання міжкультурної компетенції та використовувати цифрові інструменти, як-от Slack чи Microsoft Teams, для постійного контакту між учасниками. Ці платформи сприяють швидкому обміну інформацією та допомагають уникати непорозумінь.

Нестабільність також створює ризик нерівномірного навантаження через різницю в доступі до ресурсів і проблеми з інфраструктурою, наприклад, перебої в електропостачанні чи обмежений доступ до інтернету. Для вирішення цих проблем проджект-менеджер має забезпечити гнучкий графік роботи та впровадити чіткі процедури планування. Інструменти типу Asana чи Jira дають змогу створювати детальні плани проєктів і розподіляти задачі, щоб члени команди могли працювати автономно навіть за відсутності зв'язку. Водночас важливо забезпечити резервні комунікаційні канали, наприклад, електронну пошту чи локальні сервери для зберігання важливих даних.

Ще один важливий аспект управління в умовах нестабільності – це підтримка емоційного стану команди. Мультинаціональні колективи можуть стикатися з додатковим стресом через глобальні кризи, війни чи економічні проблеми в країнах учасників. Проджект-менеджер повинен не лише забезпечувати продуктивність, але й створювати середовище довіри, де кожен працівник відчуватиме себе почутим і підтриманим. Регулярні індивідуальні зустрічі, анонімні опитування про стан команди, а також впровадження заходів для зниження стресу (наприклад, віртуальні кава-брейки чи психологічні консультації) можуть суттєво підвищити моральний дух колективу.

До того ж у нестабільних умовах критично важливо зберігати фокус на результатах. Використання підходів, як-от OKR (Objectives and Key Results), допомагає команді зосередитися на ключових цілях, не відволікаючись на другорядні задачі. В умовах, коли ресурси можуть бути обмеженими, чітке визначення пріоритетів стає вирішальним для успішного завершення проєкту. Цифрові інструменти, як-от Google Workspace чи Confluence, сприяють централізації даних і забезпечують доступ до всієї необхідної інформації незалежно від місця перебування учасників.

В умовах цифрової економіки, що динамічно розвивається, мультинаціональні команди стають дедалі популярнішими завдяки можливості залучати фахівців із різних куточків світу. Однак ця перевага приносить і певні виклики, серед яких відмінності в часових поясах. Проджект-менеджер має враховувати ці аспекти під час планування зустрічей і термінів виконання задач. Практикою, яка дає змогу забезпечити синхронність роботи, є використання інструментів календарного планування, як-от Google Calendar, де можна відобразити часові пояси кожного члена команди. До того ж асинхронна комунікація через платформи типу Notion чи Trello сприяє тому, щоб кожен міг виконувати свою частину роботи у зручний час, не втрачаючи продуктивності.

Ще одним критичним аспектом є забезпечення ефективної передачі знань у команді. У мультинаціональних колективах часто виникає проблема зберігання й передачі інформації через мовні бар'єри чи різні рівні доступу до ресурсів. Проджект-менеджеру важливо створити єдину базу знань, де зберігаються всі документи, інструкції й матеріали, необхідні для роботи. Використання платформ для документації, як-от Confluence або SharePoint, допомагає централізувати інформацію, надаючи до неї доступ усім учасникам. Такі інструменти також сприяють стандартизації процесів, що особливо важливо в умовах нестабільності, коли комунікація може бути ускладненою.

Крім технічних аспектів, важливим складником є створення атмосфери належності до команди. У мультинаціональних колективах працівники можуть відчувати себе ізольованими через відсутність фізичного контакту або культурні відмінності. Для цього проджект-менеджер може ініціювати проведення віртуальних тимбілдингів, спільних святкувань чи навіть неформальних зустрічей у віртуальному просторі. Такі заходи допомагають покращити взаєморозуміння, формують довіру між членами команди й сприяють побудові міцніших професійних відносин. У нестабільні часи ці заходи також допомагають підтримувати мотивацію, що позитивно впливає на продуктивність проєкту.

Отже, ефективне управління мультинаціональними командами в умовах цифрової економіки потребує від проджект-менеджера адаптивності, міжкультурної обізнаності та впевненого використання цифрових інструментів. У нестабільних умовах ці навички стають ще більш актуальними, адже саме вони дають змогу забезпечити згуртованість команди, мінімізувати вплив зовнішніх факторів і досягти поставлених цілей. Успішні проджект-менеджери – це ті, хто здатен не лише координувати роботу, але й створювати середовище підтримки, яке сприяє зростанню команди навіть у найскладніших обставинах.

Список використаних джерел

1. Про цифрову трансформацію економіки в Україні. URL: https://voxukraine.org/tsyfrova-ekonomika-ukrayiny-osnovni-faktory-rozvytku (дата звертання: 10.11.2024).

2. Global Human Capital Trends by Deloitte. URL: https://www2.deloitte.com/us/en/insights/focus/human-capital-trends.html (дата звертання: 10.11.2024).

3. Harvard Business Review – Managing Remote Teams. URL: https://hbr.org/2022/10/what-great-remote-managers-do-differently (дата звертання: 10.11.2024).

4. Digital Economy Models. URL: https://www.deloitte.com/mt/en/Industries/technology/research/mt-what-is-digital-economy.html (дата звертання: 10.11.2024).

УДК 519.6

Мазур А. В., здобувачка вищої освіти,
Римар П. В., старший викладач кафедри
інформаційних технологій

ВИКОРИСТАННЯ MS EXCEL ДЛЯ РОЗВ'ЯЗАННЯ СТАТИСТИЧНИХ ЗАДАЧ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Microsoft Excel – це один із найпоширеніших інструментів для обробки даних, що активно використовується у різних галузях науки, освіти та бізнесу. Його можливості дають змогу виконувати широкий спектр статистичних задач – від базових обчислень до аналізу великих масивів даних із використанням складних формул та інструментів.

Статистичні функції [1] в Excel – це функції для аналізу значень діапазонів комірок. За допомогою статистичних функцій можна знайти найбільше і найменше значення, розрахувати середнє значення тощо.

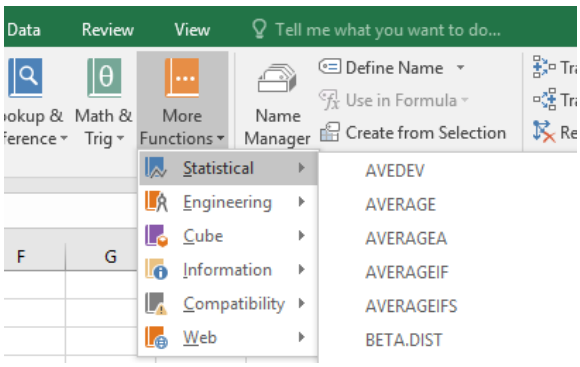


Рисунок 1 – Деякі зі статистичних функцій

Актуальність. У сучасних умовах цифрової трансформації доступ до простих і функціональних інструментів для обробки даних є дійсно важливим. Статистичний аналіз є основою у галузях науки, економіки, медицини та інших галузях. MS Excel пропонує функції, які можуть задовольнити потреби як новачків, так і професіоналів. Інтеграція з макросами VBA розширює можливості програми, даючи змогу автоматизувати рутинні процеси. Розглядаючи альтернативи, як-от SPSS, R чи Python, варто зазначити, що Excel залишається лідером завдяки зручності та інтуїтивному інтерфейсу, що робить його ідеальним для навчальних цілей і швидкого застосування у практиці.

Аналіз останніх досліджень. У багатьох наукових роботах відзначається ефективність використання MS Excel для вирішення статистичних задач:

1. Дослідження В. Петрова (2021) показало, що 75 % студентів використовують Excel для виконання лабораторних робіт зі статистики.
2. У роботі Л. Ковальчук (2020) розглянуто приклади застосування інструментів Excel для регресійного аналізу.
3. За даними М. Іваненка (2022), аналіз великих наборів даних із використанням зведених таблиць у Excel забезпечує значну економію часу, порівняно з традиційними методами.

Проте у більшості досліджень також наголошується на обмеженості Excel для роботи з дуже великими наборами даних або виконання складних статистичних моделей [2].

Мета дослідження – проаналізувати можливості MS Excel для розв’язання статистичних задач різного рівня складності, зокрема:

- виконання описової статистики;
- проведення кореляційного та регресійного аналізу.

Виклад основного матеріалу

Описова статистика. За допомогою формул AVERAGE, MEDIAN, MODE, VAR, STDEV виконано базовий аналіз. Це дало змогу визначити центральну тенденцію та варіативність даних [3].

Кореляційний аналіз. Інструмент *Data Analysis* забезпечив швидке обчислення коефіцієнта кореляції Пірсона між змінними. Результати підтвердили значний позитивний зв’язок ($r = 0{,}87$) між доходами та витратами домогосподарств.

Аналіз залежностей. MS Excel допомагає будувати графіки залежностей між змінними та знаходити трендові лінії. Наприклад, можна дослідити залежність

між доходами та витратами в домогосподарствах, використовуючи інструмент «Додати лінію тренду».

Регресійний аналіз. Проведено лінійну регресію для прогнозування витрат залежно від доходів. В Excel зручно відображати результати моделі у вигляді таблиці з коефіцієнтами та графіком. Значення $R^2 = 0{,}76$ свідчить про високу пояснювальну здатність моделі.

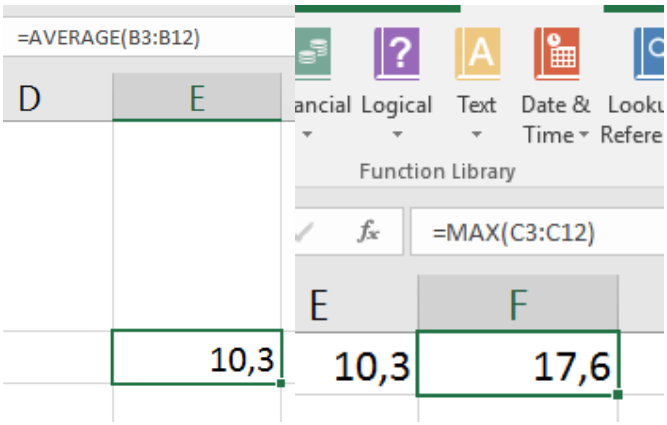


Рисунок 2 – Виконання деяких статистичних формул

	A	B	C	D
1	*X*		*Y*	
2				
3	Mean	10,3	Mean	11,6
4	Standard Error	1,11055542	Standard Error	1,175868473
5	Median	9,65	Median	12,1
6	Mode	#N/A	Mode	#N/A
7	Standard Deviation	3,51188458	Standard Deviation	3,718422605
8	Sample Variance	12,3333333	Sample Variance	13,82666667
9	Kurtosis	-1,2174592	Kurtosis	-0,893543112
10	Skewness	0,1918197	Skewness	-0,031412051
11	Range	10,4	Range	11,8
12	Minimum	5,3	Minimum	5,8
13	Maximum	15,7	Maximum	17,6
14	Sum	103	Sum	116
15	Count	10	Count	10
16	Confidence Level(95,0%)	2,51225089	Confidence Level(95,0%)	2,65999929

Рисунок 3 – Виконання операції «Correlation»

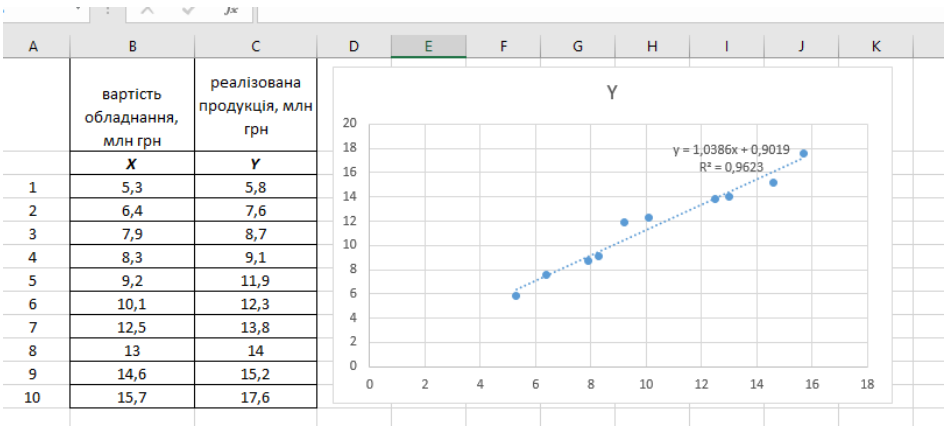
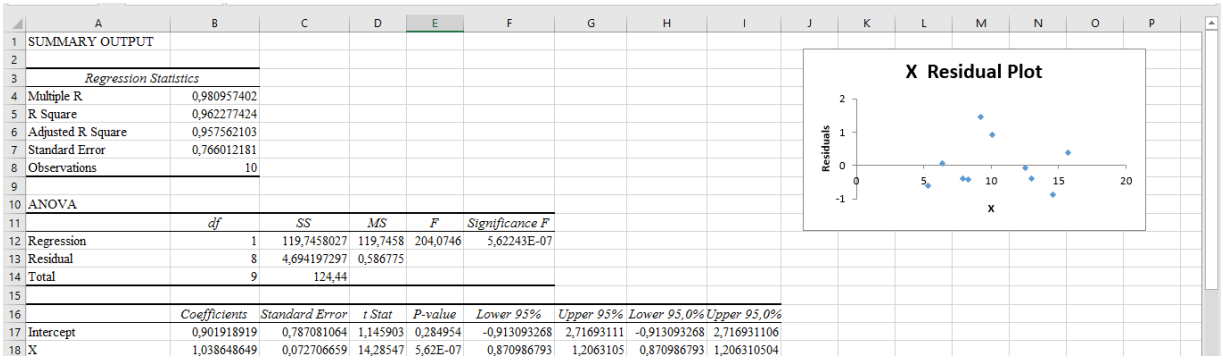


Рисунок 4 – Побудова лінії тренду



	A	B	C	D	E	F	G	H	I
1	SUMMARY OUTPUT								
2									
3	*Regression Statistics*								
4	Multiple R	0,980957402							
5	R Square	0,962277424							
6	Adjusted R Square	0,957562103							
7	Standard Error	0,766012181							
8	Observations	10							
9									
10	ANOVA								
11		*df*	*SS*	*MS*	*F*	*Significance F*			
12	Regression	1	119,7458027	119,7458	204,0746	5,62243E-07			
13	Residual	8	4,694197297	0,586775					
14	Total	9	124,44						
15									
16		*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95,0%*	*Upper 95,0%*
17	Intercept	0,901918919	0,787081064	1,145903	0,284954	-0,913093268	2,71693111	-0,913093268	2,716931106
18	X	1,038648649	0,072706659	14,28547	5,62E-07	0,870986793	1,2063105	0,870986793	1,206310504

Рисунок 5 – Проведено лінійну регресію та знайдений коефіцієнт

Висновки. MS Excel є універсальним інструментом для розв'язання широкого спектра статистичних задач завдяки простоті використання та багатофункціональності. Отже, впровадження MS Excel у навчальний процес і повсякденну роботу з даними сприяє підвищенню ефективності та доступності статистичного аналізу.

Список використаних джерел

1. Статистичні функції (довідка). URL: https://support.microsoft.com/uk-ua/excel (дата звернення: 30.11.2024).

2. Аналіз даних в Excel. URL: https://www.microsoft.com/uk-ua/microsoft-365/excel (дата звернення: 30.11.2024).

3. Талах В., Талах Т. Використання статистичних функцій Excel в аналітичних дослідженнях великих даних. *Економіка та суспільство*. 2023. Вип. 51. DOI: 10.32782/2524-0072/2023-51-15 (дата звернення: 30.11.2024).

УДК 620.9:621.316:519.87:004.942:614.83

Мельник В. Р., здобувач вищої освіти,
Якубич К. О., асистент кафедри
інформаційних технологій

МОДЕЛЮВАННЯ ОПТИМАЛЬНОГО ВІДНОВЛЕННЯ ЕНЕРГОСИСТЕМ ПІСЛЯ ПОШКОДЖЕНЬ

Донецький національний університет імені Василя Стуса, м. Вінниця

Моделювання оптимального відновлення енергосистем після пошкоджень є однією з найактуальніших проблем для України в умовах сучасної війни. Атаки на критичну інфраструктуру, зокрема на енергетичні об'єкти, значною мірою ускладнюють життя цивільного населення, створюють перешкоди для функціонування промисловості, медицини, транспорту та інших ключових секторів. Тому ефективне та швидке відновлення енергетичної системи стає не лише технічним, а й стратегічним завданням національного рівня.

Комп'ютерно-математичне моделювання відіграє ключову роль у розв'язанні цієї проблеми, оскільки дає змогу аналізувати наслідки пошкоджень,

прогнозувати їх вплив на енергосистему та планувати оптимальні шляхи відновлення. Одним із головних завдань моделювання є створення алгоритмів, що допомагають визначити пріоритетність відновлювальних робіт. Наприклад, важливо спочатку забезпечити енергопостачання для об'єктів критичної інфраструктури, як-от лікарні, водоканали, системи опалення, а також ключових промислових підприємств і транспортних вузлів.

Розробка моделей відновлення базується на комплексному аналізі даних про пошкодження, доступні ресурси та потреби населення. У цьому процесі важливими є сучасні методи, як-от оптимізація, аналіз графів, моделювання сценаріїв та імітаційні підходи. Оптимізація дає змогу знаходити найефективніші рішення для використання обмежених ресурсів, як-от ремонтна техніка, матеріали, енергетичне обладнання і людські ресурси. Використання теорії графів є важливим для аналізу структури енергомережі та виявлення найбільш уразливих її елементів.

Одним із пріоритетів відновлення енергосистеми є впровадження Smart Grid – інноваційних розумних енергомереж [1]. Це технологія, яка забезпечує автоматизацію управління, зниження втрат енергії та інтеграцію відновлюваних джерел енергії. В Україні вже реалізується пілотний проєкт у Київській області, спрямований на створення 20 000 км нових кабелів, 250 підстанцій і встановлення мільйона інтелектуальних лічильників. Ці зміни сприятимуть підвищенню стійкості мереж до пошкоджень та забезпеченню стабільного енергопостачання навіть у кризових умовах.

Іншим важливим аспектом є роль місцевих енергетичних планів (МЕПів), що розробляються за підтримки DiXi Group [2]. Вони спрямовані на сталий енергетичний розвиток громад, інтеграцію відновлюваних джерел енергії та підвищення енергоефективності. МЕПи допомагають враховувати локальні особливості, що дає змогу громадам бути автономнішими у вирішенні енергетичних проблем.

Довгострокове планування ґрунтується на Енергетичній стратегії України до 2050 р., яка передбачає кліматичну нейтральність, скорочення використання вугілля, модернізацію енергетичної інфраструктури та інтеграцію з ринками ЄС [3]. Ця стратегія акцентує на розвитку альтернативних джерел енергії та інновацій, що зробить енергосистему більш гнучкою і безпечною.

Моделювання оптимального відновлення енергосистем має не тільки вирішувати поточні проблеми, але й закладати основи для майбутньої стійкості та модернізації енергетики України. Важливо не лише інтегрувати новітні технології, як-от Smart Grid, але й максимально залучати локальні громади до процесу планування через розробку МЕПів.

Для підвищення ефективності відновлення автори пропонують створити єдину національну цифрову платформу для моніторингу стану енергосистеми та обміну даними між усіма учасниками – від ремонтних бригад до органів місцевого самоврядування. Ця платформа могла б забезпечити оперативну координацію дій, прозорість розподілу ресурсів та інтеграцію міжнародної допомоги.

Підхід до відновлення має бути системним і довгостроковим. Це не лише питання про те, як повернути енергосистему до робочого стану, але й як зро-

бити її надійнішою, сучаснішою та адаптованою до майбутніх викликів. Ефективне моделювання, спільна робота держави, громад і міжнародних партнерів допоможе Україні не тільки відновитися, але й стати прикладом інноваційного підходу до енергетичної трансформації.

Список використаних джерел

1. Smart Grid в Україні: що це таке, навіщо потрібне і коли з'явиться. URL: https://mind.ua/publications/20259406-smart-grid-v-ukrayini-shcho-ce-take-navishcho-potribne-i-koli-z-yavitsya (дата звернення: 30.11.2024).

2. Як покращити Методику місцевих енергетичних планів: DiXi Group підбив проміжні підсумки проєкту з розробки МЕПів для громад. URL: https://dixigroup.org/yak-pokrashyty-metodyky-miszevych-energetychnyh-planiv-dixi-group-pidbyv-promizhni-pidsumky-proekty-z-rozrobky-mepiv-dlia-gromad/ (дата звернення: 30.11.2024).

3. Енергетичної стратегії України до 2050 р. URL: https://mev.gov.ua/reforma/enerhetychna-stratehiya-0 (дата звернення: 30.11.2024).

УДК 004.94

Михайляк М. О., здобувачка вищої освіти,
Комаров В. Ф., канд. техн. наук,
старший викладач кафедри прикладної
математики та кібербезпеки

МОДЕЛЮВАННЯ ФІЗИЧНИХ СИСТЕМ ЗАСОБАМИ MAPLE ТА SIMULINK

Донецький національний університет імені Василя Стуса, м. Вінниця

Моделювання реальних систем є невід'ємною частиною сучасного науково-технічного прогресу. Це дає змогу вивчати поведінку складних об'єктів і процесів без дорогих експериментів та прогнозувати результати змін, оптимізувати параметри і підвищувати ефективність систем.

Ключові завдання моделювання включають аналіз для розуміння принципів функціонування системи та її характеристик, прогнозування, яке допомагає зрозуміти те, як система поводитиметься за різних умов, оптимізацію для визначення найкращих рішень для досягнення поставлених цілей, а також візуалізацію для представлення складних процесів у вигляді графічних матеріалів, що полегшують прийняття управлінських рішень.

Maple – це потужна система комп'ютерної алгебри (Computer Algebra System, CAS), яку можна використовувати для вирішення широкого кола математичних задач. Вона може розв'язувати рівняння аналітично, виконувати інтегрування, диференціювання та інші символьні обчислення, які важливі в багатьох галузях науки та техніки [1].

Maple має низку ключових переваг, які роблять її потужним інструментом для розв'язання математичних задач і моделювання. Одна з головних переваг – підтримка символьних розрахунків, що дає змогу працювати з загальними формулами та аналітичними виразами, не обмежуючись лише числовими да-

ними. Це значно розширює можливості аналізу й дає змогу отримувати точні теоретичні рішення. До того ж Maple забезпечує візуалізацію даних у вигляді графіків, схем і тривимірних моделей, що полегшує розуміння результатів і їх презентацію [2].

Ще однією перевагою є інтеграція з іншими програмними засобами, як-от MATLAB, Python і R, а також підтримка стандартних форматів обміну даними. Це сприяє гнучкості використання Maple у різних проєктах. Завдяки функціям програмування Maple допомагає автоматизувати складні обчислення, що заощаджує час і знижує ймовірність помилок, роблячи її ефективним інструментом для наукових та інженерних завдань. Отже, Maple є універсальним інструментом, що поєднує точність, гнучкість і зручність для вирішення математичних задач.

SIMULINK – це середовище для моделювання динамічних систем, яке дає змогу створювати та аналізувати моделі за допомогою блочного представлення [3]. Ця платформа широко використовується для розробки систем управління, обробки сигналів, моделювання фізичних процесів та інших завдань, що вимагають динамічного підходу. Особливістю SIMULINK є можливість виконання моделювання в реальному часі.

Основні можливості включають моделювання систем управління, обробку сигналів і фізичних процесів, а також створення моделей для різних технічних застосувань. Платформа підтримує інтеграцію з MATLAB для виконання більш складного аналізу та обчислень, дає змогу отримувати точніші результати.

Переваги SIMULINK полягають у інтуїтивно зрозумілому інтерфейсі, який допомагає швидко створювати моделі, великому виборі готових блоків для моделювання різноманітних систем.

Якщо порівнювати Maple і SIMULINK, основна різниця полягає в їх підході до моделювання та сферах застосування. Maple більше підходить для побудови математичних моделей фізичних процесів та вирішення диференціальних рівнянь SIMULINK. Натомість орієнтований на динамічне моделювання та інтерактивну перевірку систем. Він дає змогу проводити симуляції в реальному часі, миттєво змінювати параметри системи та оцінювати її поведінку в різних умовах. Це робить його незамінним для розробки практичних застосувань, як-от системи управління, оптимізація сигналів чи тестування прототипів.

Інтеграція цих двох інструментів відкриває нові можливості для комплексного моделювання, об’єднуючи переваги аналітичного підходу з можливостями динамічної симуляції. Використовуючи Maple, можна розрахувати необхідні значення чи параметри для моделювання. Ці результати легко передаються в SIMULINK, де моделі перевіряються на практиці та піддаються тестуванню у змінних умовах.

Такий підхід не лише скорочує час на розробку, але й підвищує надійність систем, оскільки кожен етап, від аналітичного аналізу до симуляції, інтегрується в єдиний робочий процес. Це особливо корисно в проєктах, що вимагають точності й гнучкості, як-от інженерні системи чи розробка алгоритмів управління.

Ці платформи використовуються для вирішення широкого спектра завдань у різних галузях. У фізиці їх застосовують для дослідження явищ, як-от теплопровідність чи динаміка механічних конструкцій, що вимагають точного

моделювання складних процесів. У техніці вони допомагають розробляти системи управління, наприклад, проєктувати електричні схеми або створювати алгоритми для автоматизації роботів.

В економіці ці засоби застосовуються для побудови моделей, що аналізують фінансові потоки, або для оптимізації розподілу ресурсів у великих організаціях. Отже, Maple і SIMULINK забезпечують ефективний інструментарій для розв'язання спеціалізованих задач у науці, техніці та бізнесі.

Maple і SIMULINK забезпечують потужні можливості для моделювання, аналізу та вдосконалення складних систем у різних галузях. Залежно від характеру завдання Maple підходить для точних розрахунків і теоретичного обґрунтування, тоді як SIMULINK ідеально підходить для динамічного моделювання та перевірки поведінки систем у реальних умовах. Їх поєднання дає змогу досягти балансу між глибоким аналітичним підходом і практичною реалізацією, що робить ці платформи універсальними для інженерних і наукових досліджень.

Перспективи розвитку Maple та SIMULINK зосереджуються на покращенні їх взаємодії, що допоможе автоматизувати більш складні процеси моделювання та зменшити час на налаштування моделей. Очікується розширення бібліотек SIMULINK, що дасть змогу створювати ще більш спеціалізовані блоки для широкого спектра галузей, як-от аерокосмічна або біомедична інженерія. До того ж розробка нових алгоритмів для аналізу результатів симуляції допоможе спростити інтерпретацію даних і зробити процес оптимізації моделей швидшим і точнішим.

Список використаних джерел

1. Використання математичного пакету Maple для розв'язування та моделювання задач. URL: https://lib.pnu.edu.ua:8080/bitstream/123456789/3286/1/Використання математичного пакету maple для розв'язування та моделювання задач_методичка.pdf (дата звернення: 02.12.2024).

2. Ніколюк П. К. Моделювання систем: навч. посіб. для здобувачів вищої освіти спеціальності 122 «Комп'ютерні науки». Вінниця: ДонНУ, 2023. 33 с. (дата звернення: 02.12.2024).

3. Introduction: Simulink Modeling. URL: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SimulinkModeling (дата звернення: 02.12.2024).

УДК 004.43:004.774.6

Мишківська Я. В., здобувачка вищої освіти,
Січко Т. В., канд. техн. наук, доцент,
доцент кафедри інформаційних технологій

АНАЛІЗ ПРОДУКТИВНОСТІ ХУКІВ У ПОРІВНЯННІ З КЛАСОВИМИ КОМПОНЕНТАМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Користувацький інтерфейс (UI) є критично важливим для залучення та утримання користувачів. Інтуїтивність, адаптивність і привабливий дизайн за-

безпечують позитивний досвід взаємодії з вебзастосунком. React (інколи React.js, ReactJS) – це JavaScript-бібліотека для створення гнучких та сучасних користувацьких інтерфейсів (UI) для вебзастосунків. Вона дає змогу розробити все те, з чим користувач вебресурсу може взаємодіяти напряму: привабливе оформлення сайта, ефектні анімації, адаптивний дизайн, який підлаштовується під різні пристрої, тощо [1]. Впровадження хуків стало переломним моментом у розвитку React, оскільки вони значно спростили управління станом і побічними ефектами у функціональних компонентах, підвищивши ефективність розробки. Цей перехід актуалізував питання про вибір між класовими та функціональними компонентами залежно від вимог проєкту.

До появи хуків класовий підхід був основним способом розробки в React. Він базується на ES6-класах і використовує методи життєвого циклу для управління станом та побічними ефектами. Особливості класових компонентів у React включають використання this.state для зберігання стану, методів життєвого циклу для управління логікою та необхідність прив'язки контексту this для методів.

Хуки – це функції, за допомогою яких можна «зачепитися» за стан та методи життєвого циклу React із функційних компонентів [2]. Найцікавіша їх особливість полягає в тому, що їх можна викликати лише на найвищому рівні компонента. На практиці це означає, що не можна викликати хуки всередині умов (if), циклів, інших функцій або після раннього return. Це правило забезпечує виклик хуків в одному і тому ж порядку під час кожного ререндера компонента [3].

У табл. 1 наведено порівняльний аналіз кожного підходу.

Таблиця 1 – Порівняння хуків і класових компонентів

Критерій	**React Hooks (Функціональні компоненти)**	**Класові компоненти**
Час рендерингу	• Швидший завдяки відсутності методів життєвого циклу. • Оптимізація через мемоізацію (React.memo, useCallback, useMemo)	• Повільніший через обробку методів життєвого циклу
Споживання пам'яті	• Економніше: не створюються екземпляри класів	• Більше витрат пам'яті через створення екземплярів класів
Читабельність коду	• Код короткий, декларативний, зосереджений на функціях	• Громіздкий код із численними методами життєвого циклу
Обробка побічних ефектів	• Використання useEffect, що дає змогу гнучко контролювати залежності	• Використання методів життєвого циклу (componentDidMount, componentDidUpdate)
Гнучкість повторного використання логіки	• Легко реалізується через кастомні хуки	• Повторне використання складніше: потрібні HOC або Render Props
Продуктивність у великих додатках	• Висока за умови правильної мемоізації залежностей	• Може знижуватися через складну логіку в методах життєвого циклу
Підтримка тестування	• Зручно тестувати завдяки ізольованим функціональним підходам	• Тестування складніше через наявність стану та контексту класів

Отже, хуки забезпечують вищу продуктивність, економію ресурсів і спрощення коду, що робить їх ідеальним вибором для сучасних проєктів. Вони спрощують управління станом і побічними ефектами, даючи змогу зосередитися на логіці компонента. Класові компоненти залишаються доцільними для підтримки старих систем із наявною логікою, але їх поступовий перехід на хуки підвищить ефективність розробки.

Список використаних джерел

1. Як стати React розробником. Що потрібно знати та вміти – з нуля до рівня спеціаліста. ITVDN. URL: https://itvdn.com/ua/blog/article/how-to-become-a-react-developer (дата звернення: 26.11.2024).

2. Огляд хуків. React Documentation. URL: https://uk.legacy.reactjs.org/docs/hooks-overview.html (дата звернення: 26.11.2024).

3. Розбираємося з хуками в React. DOU. URL: https://dou.ua/forums/topic/47858/ (дата звернення: 28.11.2024).

4. Павлов Д. Л., Січко Т. В. Принцип роботи Web API та його застосування. Прикладні інформаційні технології: матеріали всеукр. наук.-практ. конф. (м. Вінниця, 2023). С. 166–168.

УДК 004.652

Морозюк А. А., здобувач вищої освіти,
Зелінська О. В., канд. техн. наук, доцент,
доцент кафедри інформаційних технологій

ІНТЕГРАЦІЯ ПРИНЦИПІВ ПРОДУКТОВОЇ АНАЛІТИКИ У ДИЗАЙНІ ІНФОРМАЦІЙНИХ СИСТЕМ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні інформаційні системи відіграють ключову роль у нашому житті, забезпечуючи функціональність і підтримуючи роботу різних сфер – від бізнесу до охорони здоров’я. Однак часто такі системи не відповідають очікуванням користувачів, оскільки дизайнери ігнорують важливість аналізу поведінки аудиторії та розуміння її реальних потреб. Як наслідок, це призводить до низького рівня залучення, проблем з інтуїтивністю інтерфейсу та навіть відмови від використання продукту. Саме тому інтеграція підходів продуктової аналітики у дизайн інформаційних систем набуває все більшої актуальності. Використання даних про поведінку користувачів та їхні вподобання дає змогу створювати продукти, які не лише виконують свої функції, а й приносять задоволення від взаємодії. Поєднання даних та дизайнерського мислення допомагає уникнути розриву між тим, чого потребують користувачі, і тим, що пропонують розробники.

Метою цієї роботи є опис основних принципів продуктової аналітики та аналіз способів їх впровадження у процес проєктування інформаційних систем. Під час дослідження будуть розглянуті ключові аспекти інтеграції аналітичних даних у дизайн, а також приклади їх використання для створення більш ефективних продуктів.

Продуктова аналітика – це процес збору, обробки та інтерпретації даних про взаємодію користувачів із продуктом. Вона допомагає командам зрозуміти, як клієнти використовують продукт, які функції є найбільш популярними, а також виявити проблеми у взаємодії. Завдяки продуктовій аналітиці розробники й дизайнери можуть приймати обґрунтовані рішення для покращення користувацького досвіду [1].

Є кілька показників, без яких неможливо уявити ефективний аналіз:

1. Залучення користувачів – це про те, наскільки активно люди користуються продуктом. Наприклад, якщо в застосунку є функція, яка майже не використовується, це може бути сигналом для змін. Ця метрика показує, що працює добре, а що потребує покращення [2, 3].

2. Утримання – це як відповідь на запитання: чи повертаються користувачі до продукту? Наприклад, якщо користувачі активно встановлюють застосунок, але через тиждень видаляють його, потрібно зрозуміти, у чому проблема – можливо, інтерфейс заплутаний або функціонал не відповідає очікуванням.

3. Вартість залучення клієнтів (CAC) та життєва цінність клієнта (CLV) – показники, які важливі не лише для бізнесу, а й для дизайну. Якщо створити інтуїтивний і привабливий інтерфейс, користувачі будуть активніше повертатися, а це підвищує CLV і знижує витрати на залучення нових клієнтів [3].

Методи збору даних – це окремий, доволі важливий аспект. Наприклад:

1. A/B тестування дає змогу порівнювати два варіанти дизайну й обирати той, що краще працює. Це як експеримент, де видно, який підхід подобається користувачам [2].

2. Теплові карти (heatmaps) – це спосіб «бачити» дії користувачів. Наприклад, можна зрозуміти, на які кнопки вони натискають найчастіше, а які ігнорують [2].

3. Аналіз сесій дає можливість «стежити» за тим, як користувач рухається по продукту. Це допомагає знайти моменти, де людина може заплутатися або втратити інтерес [1].

Наведені метрики та методи – основа для будь-якого продукту. Вони допомагають зрозуміти, чи йде продукт у правильному напрямі, і роблять дизайн не лише красивим, але й корисним.

Для того, щоб ефективно застосувати ці метрики та методи в реальному житті, важливо інтегрувати їх у процес проєктування інформаційних систем. Саме аналітика допомагає не лише оцінювати продукт, але й безпосередньо впливати на його дизайн. Це забезпечує створення таких рішень, які одночасно відповідають очікуванням користувачів і є максимально зручними в користуванні. Інтеграція аналітичних підходів охоплює кілька ключових етапів, кожен із яких має важливе значення для покращення якості продукту:

1. Перший етап – збір і аналіз даних. Основним його завданням є збирання інформації про взаємодію користувачів із продуктом. Використовуються такі методи: теплові карти, аналіз сесій та поведінкові дослідження. Це дає змогу визначити найбільш затребувані функції системи, а також виявити проблемні ділянки в користувацькому досвіді [1, 2].

2. Наступний етап – формування вимог до дизайну на основі аналітичних даних. Наприклад, якщо дані показують, що певний елемент інтерфейсу не

виконує своїх функцій через низьке залучення користувачів, його структура чи розташування потребує оптимізації [3, 4].

3. Останнє – тестування прототипів з реальними даними. На цьому етапі створюються прототипи, які тестуються з використанням даних, зібраних під час аналізу. Це допомагає перевірити гіпотези про покращення дизайну та виявити можливі недоліки ще до впровадження змін у продукт [3].

Приклади покращення дизайну за допомогою аналітики:

1. Оптимізація функціоналу системи на основі даних про активність користувачів дає змогу визначити найпопулярніші та найменш затребувані функції. Наприклад, якщо статистика показує, що користувачі часто відкривають певну сторінку, але не завершують дію (до прикладу, оформлення замовлення чи заповнення форми), це може вказувати на необхідність спрощення процесу або додавання інструкцій [1, 2].

2. Ретельний аналіз точок, де користувачі залишають продукт, допомагає виявити проблемні аспекти інтерфейсу, як-от перевантажені сторінки чи складнощі в розумінні функціоналу. Наприклад, якщо значна частина користувачів покидає сайт на етапі реєстрації, це може вказувати на потребу спростити форму чи переглянути її структуру для полегшення введення даних [3].

3. Аналіз поведінки користувачів може виявити труднощі у навігації системою. Наприклад, якщо теплові карти показують, що користувачі рідко переходять на важливу сторінку або витрачають багато часу на пошук потрібної функції, це може сигналізувати про незрозумілу структуру меню чи невдале розташування кнопок. На основі цих даних можна змінити структуру меню або зробити ключові елементи інтерфейсу більш видимими та доступними [2].

Використання аналітики допомагає виявити слабкі місця дизайну та оптимізувати його для покращення взаємодії з користувачами. Аналіз активності, точок відтоку та навігаційних труднощів дає змогу зробити продукт зручнішим і зрозумілішим, що сприяє підвищенню ефективності роботи системи та задоволеності користувачів.

Отже, інтеграція принципів продуктової аналітики у дизайн інформаційних систем є важливим кроком до створення ефективних, зручних і користувацько-орієнтованих продуктів. Використання даних про поведінку дає змогу аудиторії дизайнерам і розробникам не лише ідентифікувати проблеми, але й знаходити оптимальні шляхи їх вирішення. Поєднання аналітичних підходів із дизайнерським мисленням забезпечує створення інформаційних систем, які відповідають високим стандартам якості й успішно задовольняють очікування користувачів.

Список використаних джерел

1. Design in Analytics: A Unique Intersection Driving Innovation and Value. URL: https://cutt.ly/eeLPYKI1 (дата звернення: 20.11.2024).

2. Analytics design: what it is? How is it crucial for your websites strategies?. URL: https://cutt.ly/IeLPUvxC (дата звернення: 20.11.2024).

3. Data-driven підхід у продуктовому дизайні. URL: https://cutt.ly/OeLPUS0D (дата звернення: 22.11.2024).

4. The Influence of Data Analytics on Design. URL: https://cutt.ly/YeLPU38s (дата звернення: 26.11.2024).

5. Як мінімізувати ризик розробки неправильного рішення: збір даних. URL: https://cutt.ly/heLPIlFA (дата звернення: 27.11.2024).

УДК 004.94

Морозюк А. А., здобувач вищої освіти,
Комаров В. Ф., канд. техн. наук,
старший викладач кафедри прикладної
математики та кібербезпеки

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЯК ІНСТРУМЕНТ БОРОТЬБИ ІЗ ТРАНСПОРТНИМИ ЗАТОРАМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасний світ стикається зі значним зростанням кількості транспортних засобів, що призводить до підвищення навантаження на дорожню інфраструктуру. Однією з найгостріших проблем є виникнення заторів, які впливають на різні аспекти життєдіяльності людини та функціонування міст. Ефективне вирішення цієї проблеми є ключовим завданням транспортного планування і суспільства загалом.

Затори завдають суттєвої шкоди економіці, екології та якості життя. У економічному аспекті вони спричиняють значні фінансові втрати через збільшення витрат на паливо, зниження продуктивності та затримки у перевезеннях вантажів і пасажирів. За оцінками, економічні втрати від заторів у великих містах можуть сягати мільярдів доларів щорічно [1, 2].

Екологічні наслідки також є значними: у заторах транспорт працює у режимі зупинок та простою з працюючими двигунами, що збільшує рівень викидів вуглекислого газу та інших шкідливих речовин. Це погіршує якість повітря та впливає на здоров'я населення великих міст [1]. До того ж затори змушують марно витрачати час – щорічно десятки годин очікування у дорожньому русі, що могли бути використані для продуктивної діяльності чи відпочинку [2].

Комп'ютерне моделювання є одним із найефективніших інструментів для аналізу та вирішення транспортних проблем. Його основа – це математичний опис потоків транспорту з урахуванням фізичних, поведінкових та технічних факторів. Процес моделювання починається зі збору та аналізу даних: кількість транспортних засобів, швидкість руху, стан дорожньої інфраструктури та поведінкові особливості водіїв. На основі цих даних створюється математична модель, яка дає змогу точно відобразити реальну транспортну систему, що є ключем до проведення точних симуляцій і тестування різних сценаріїв. Результати таких симуляцій забезпечують прийняття оптимальних рішень, спрямованих на підвищення ефективності транспортних систем [1].

Особливу роль у цьому процесі відіграє програмне забезпечення, зокрема AnyLogic North America LLC, яке є універсальним інструментом для аналізу та оптимізації транспортних систем. Цей інструмент допомагає досліджувати

проблемні ділянки дорожньої мережі та вдосконалювати алгоритми управління світлофорами. Це сприяє покращенню регулювання трафіка й оптимізації інфраструктурних змін, як-от будівництво нових розв'язок або оптимізація маршрутів громадського транспорту. Отже, AnyLogic виступає важливим інструментом для прогнозування наслідків транспортних рішень [4].

Фізика трафіка також є важливим компонентом транспортного моделювання, адже вона враховує ключові показники, як-от швидкість, щільність і потік транспортних засобів. Для моделювання транспортних процесів використовуються три основні підходи: макроскопічний, мікроскопічний і мезоскопічний. Макроскопічні моделі оцінюють транспортні потоки загалом, подібно до рідини. Мікроскопічні моделі, навпаки, аналізують поведінку окремих транспортних засобів, враховуючи, наприклад, дистанцію між автомобілями чи їх реакцію на зміну умов руху. Мезоскопічні моделі поєднують обидва підходи, даючи змогу вивчати взаємодії транспортних груп у реальному часі. Завдяки цьому моделюванню фахівці отримують детальну картину динаміки транспортних потоків [1].

Одним із цікавих сценаріїв для дослідження є так звані фантомні затори – явища, що виникають через хаотичну поведінку водіїв навіть за відсутності очевидних перешкод. Комп'ютерне моделювання дає змогу глибше зрозуміти природу цих явищ і розробити стратегії для їх мінімізації. Наприклад, моделювання може використовуватись для оцінки впливу автономних транспортних засобів на зменшення таких заторів, зокрема у взаємодії зі звичайними автомобілями. Також моделі застосовуються для перевірки ефективності змін у дорожній інфраструктурі, що дає змогу заздалегідь оцінити їх вплив на транспортні потоки [2, 3].

Отже, комп'ютерне моделювання виступає потужним інструментом для дослідження транспортних систем. Воно дає змогу розробляти та впроваджувати рішення, спрямовані на зменшення заторів, підвищення безпеки дорожнього руху та поліпшення загальної ефективності транспортних систем. Завдяки точним симуляціям і аналізу різних сценаріїв можна прогнозувати наслідки змін у транспортній інфраструктурі та знаходити оптимальні стратегії управління трафіком. Це сприяє створенню більш комфортних та екологічних умов для всіх громадян.

Список використаних джерел

1. Transportation Modelling: Challenges & Solutions. URL: https://www.ptvgroup.com/en/application-areas/transportation-modeling (дата звернення: 29.11.2024).

2. What is a phantom traffic jam, why do they happen and how can they be stopped? URL: https://www.jurnileasing.co.uk/blog/what-is-a-phantom-traffic-jam-why-do-they-happen-and-how-can-they-be-stopped (дата звернення: 29.11.2024).

3. The physics of traffic. URL: https://www.here.com/learn/blog/the-physics-of-traffic (дата звернення: 30.11.2024).

4. Simulation software for Every Business Challenge. URL: https://www.anylogic.com/features/ (дата звернення: 30.11.2024).

УДК 004.72

Наральник Б. Ю., здобувач вищої освіти,
Ніколюк П. К., д-р фіз.-мат. наук,
професор, професор кафедри
комп'ютерних технологій

ОБРОБКА ДАНИХ У ХМАРНИХ СЕРЕДОВИЩАХ: ВИКЛИКИ ТА ПЕРСПЕКТИВИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Сучасна цифрова епоха супроводжується вибуховим зростанням обсягів даних, які необхідно зберігати, обробляти та аналізувати. Згідно зі звітами провідних аналітичних компаній, до 2025 р. обсяг глобального цифрового контенту сягне понад 180 зетабайтів, що створює нові виклики для традиційних систем зберігання та обробки даних [1]. У цьому контексті хмарні технології виступають одним із ключових інструментів, які дають змогу ефективно вирішувати завдання масштабування, продуктивності та економічної ефективності.

Актуальність. Багато організацій, включно з міжнародними корпораціями, державними установами та стартапами, переходять на хмарні платформи, як-от AWS, Google Cloud чи Azure, щоб скоротити витрати на локальну інфраструктуру та забезпечити гнучкість своїх операцій.

У зв'язку з необхідністю обробляти неструктуровані дані, як-от відео, текст і дані IoT, традиційні підходи більше не відповідають сучасним потребам. Хмарні платформи забезпечують адаптивність та високу обчислювальну потужність для аналізу таких даних у реальному часі.

Хмарні рішення слугують базою для розвитку технологій штучного інтелекту (AI), машинного навчання (ML), великих даних (Big Data) та Інтернету речей (IoT). Вони допомагають об'єднувати та обробляти дані з різних джерел, створюючи нові можливості для бізнесу та науки.

Хмарні технології не лише змінюють підходи до обробки даних, а й сприяють розвитку економіки, відкриваючи нові робочі місця та вдосконалюючи якість послуг. Вони дають змогу компаніям будь-якого масштабу конкурувати на глобальному ринку, використовуючи передові інструменти аналізу даних.

Основні виклики обробки даних у хмарних середовищах. Хмарні платформи стають мішенню для кіберзлочинців, оскільки в них зберігаються великі обсяги чутливої інформації. У разі порушення безпеки зловмисники можуть отримати доступ до конфіденційної інформації, включно з персональними даними користувачів, фінансовою інформацією або корпоративними секретами.

Атаки типу DDoS можуть паралізувати роботу хмарних сервісів, створюючи перебої у доступі до даних. Працівники компаній чи адміністратори хмарної платформи можуть випадково чи навмисно спричинити витік даних.

Захист даних потребує впровадження багаторівневої безпеки, використання шифрування, контроль доступу та регулярний аудит системи.

Одним із головних викликів є затримки у передачі даних між клієнтом і хмарним центром обробки. Це особливо критично для систем реального часу,

як-от відеоконференції, онлайн-ігри або системи моніторингу IoT. Причини цього такі:

- обмеження пропускної здатності мережі (швидкість передачі даних може бути недостатньою для великих обсягів інформації);
- віддаленість дата-центрів (чим далі розташований дата-центр, тим більший час потрібен для передачі даних);
- залежність від мережевого обладнання (нестабільність чи застарілість мережевої інфраструктури можуть суттєво вплинути на продуктивність).

Для вирішення цих проблем впроваджуються технології оптимізації трафіка, розподілу даних між кількома дата-центрами та використання локальних гібридних хмар.

Незважаючи на те, що хмарні платформи дають змогу знизити початкові витрати, у довгостроковій перспективі можуть виникати значні витрати. Складні операції обробки, особливо аналітичні задачі, можуть бути дорогими через високу потужність обчислювальних ресурсів.

Оптимізація витрат включає правильний вибір тарифного плану, використання автоматизованих інструментів для моніторингу витрат і аналізу ефективності використання ресурсів.

Висновки. Впровадження квантових обчислень та вдосконалених алгоритмів дають змогу значно прискорити обробку великих обсягів даних. Інтеграція штучного інтелекту необхідна для моніторингу загроз і блокчейн-технологій для забезпечення прозорості та безпеки операцій з даними.

Поєднання публічних та приватних хмар забезпечить гнучкість у керуванні даними, балансуючи між доступністю та конфіденційністю. Використання енергоефективних дата-центрів і технологій зменшення енергоспоживання сприятиме зниженню впливу на довкілля.

Список використаних джерел

1. Volume of data / information created. URL: https://www.statista.com/statistics/871513/worldwide-data-created/ (дата звернення: 30.11.2024).

2. Що таке хмарні технології і їх використання. URL: https://biznestrendy.com.ua/shcho-take-khmarni-tekhnolohii-i-ikh-vykorystannia/ (дата звернення: 30.11.2024).

3. Хмарні технології 2024 – що це таке та які хмари найкращі? URL: https://ucloud.ua/hmarni-tehnologiyi-shho-cze-take/ (дата звернення: 30.11.2024).

УДК 004.94:519.6

Підруцький Д. А., здобувач вищої освіти,
Комаров В. Ф., канд. техн. наук,
старший викладач кафедри прикладної
математики та кібербезпеки

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЕКОЛОГІЧНИХ СИСТЕМ, ЩО ЗАЗНАЛИ ШКОДИ ПІД ЧАС ВІЙНИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні воєнні конфлікти спричиняють масштабні екологічні катастрофи, зокрема руйнування критичної інфраструктури, як-от дамби, водосховища, промислові об'єкти та енергетичні станції. Це призводить до порушення екосистем, затоплення територій, забруднення водних ресурсів і втрати біорізноманіття. Комп'ютерне моделювання виступає ключовим інструментом для аналізу, прогнозування та розробки заходів із ліквідації наслідків таких катастроф.

На жаль, через жахливі речі, а саме війну, доволі часто страждає і екологія. Особливо коли руйнації зазнають безпрецедентно важливі об'єкти, такі як дамби чи водосховища. Їх руйнування має низку таких критичних ризиків [1]:

- Затоплення територій. Вода може охоплювати значні площі, руйнуючи інфраструктуру, сільськогосподарські угіддя та оселі.
- Порушення водопостачання. Зниження рівня води у водосховищах впливає на водопостачання міст, сіл і промислових об'єктів.
- Екологічні наслідки. Руйнування екосистем через вимивання ґрунтів, загибель флори і фауни, забруднення води хімічними речовинами.

Задля запобігання такому часто застосовують моделювання катастроф для того, щоб краще розуміти імовірні наслідки [2]. В таких випадках комп'ютерне моделювання є дуже доречним, оскільки дає змогу:

- оцінити масштаби наслідків на основі даних про топографію, швидкість і об'єм виливу води (можна спрогнозувати зону затоплення);
- розробити стратегії реагування завдяки тому, що моделювання допомагає оцінити ефективність інженерних рішень, як-от будівництво захисних споруд, евакуація та відновлення екосистем;
- аналізувати довгострокові впливи, а особливо наслідки, через моделі для водних ресурсів і екології регіону на десятиліття вперед.

Для моделювання зруйнованих дамб і водосховищ використовуються спеціалізовані інструменти, серед яких є спеціальна програма HEC-RAS (Hydrologic Engineering Center's River Analysis System) [3]. Це сучасне програмне забезпечення, розроблене Інженерним центром гідрології США, яке застосовується для моделювання руху води у відкритих каналах, заплавах та інженерних спорудах. Його основне призначення полягає у виконанні розрахунків профілів водної поверхні для постійних і змінних потоків, моделюванні переносу осадів і оцінці якості води в різних умовах.

Ця програма використовується для управління водними ресурсами, аналізу зон затоплення, оцінки ризиків повеней і проєктування захисних споруд. Вона

дає змогу оцінювати ефективність роботи дамб, шлюзів, мостів та інших об'єктів, аналізувати наслідки ерозії, акумуляції осадів і змін якості води. Застосування HEC-RAS дає змогу створювати детальні математичні моделі для відображення потоків води, визначати оптимальні стратегії управління водними системами, планувати заходи зі зменшення затоплення та прогнозувати вплив інженерних рішень на навколишнє середовище.

Програма активно використовується у сфері муніципального управління для аналізу інфраструктури, запобігання затопленням і планування водопостачання, а також в екологічних дослідженнях, які стосуються оцінки впливу змін у водних системах на екосистеми. Завдяки розвинутому інтерфейсу HEC-RAS допомагає візуалізувати результати моделювання, що є важливим для прийняття ефективних рішень на основі отриманих даних.

Отже, комп'ютерне моделювання виступає незамінним інструментом для аналізу та управління наслідками руйнування екологічних систем, спричинених воєнними діями. Воно дає змогу оцінювати масштаби катастроф, прогнозувати ризики затоплення та забруднення, а також розробляти ефективні стратегії реагування. Завдяки детальним симуляціям можна точно визначати зони ризику, оцінювати вплив інженерних рішень і заздалегідь планувати вчасні заходи для мінімізації шкоди екосистемам. Це сприяє більш ефективному управлінню водними ресурсами та відновленню природного балансу в умовах екологічних криз.

Список використаних джерел

1. Системний підхід до оцінки ризиків надзвичайних ситуацій в Україні / В. Д. Калугін, В. В. Тютюник, Л. Ф. Чорногор, Р. І. Шевченко. *Східноєвропейський журнал передових технологій*. 2012. № 1/6(55). Харків. С. 59–70.

2. Черенкевич О. С. Статистичне моделювання екологічних ризиків як фактора екологічної безпеки. *Статистика України.* 2020. № 2–3. С. 59–67. DOI: 10.31767/ su.2-3(89-90) 2020.02-03.07.

3. Яке призначення HEC-RAS? URL: https://bioglobin.com.ua/ukraincyam/yake-priznachennya-hec-ras.html (дата звернення: 04.12.2024).

УДК 004.422.5.

Синіцький В. Ю., здобувач вищої освіти,
Поремський Ю. В., канд. техн. наук, старший викладач кафедри інформаційних технологій

РЕКУРСИВНІ АЛГОРИТМИ В СУЧАСНОМУ ПРОГРАМУВАННІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Рекурсія – це метод розв'язання задачі шляхом розбиття її на підзадачі, що є спрощеними варіантами тієї ж задачі. Ключовою ідеєю цього методу є поетапне зведення задачі до її найпростішої версії, що має кінцеве рішення. Найпростішу версію називають базовим випадком. Базовий випадок є обов'язковим елементом у рекурсивних алгоритмах. Як приклад рекурсії можна навести обчислення чисел Фібоначчі, де кожне наступне число є сумою двох попер-

ніх. Такий алгоритм обчислення яскраво відображає те, як рекурсія допомагає будувати рішення подібних задач [1].

Існує декілька форм рекурсії, розглянемо деякі з них. Хвостова рекурсія є формою рекурсивного алгоритму, в якому остання дія, виконувана у функції перед поверненням значення, – її власний рекурсивний виклик. Такий підхід використовується для ефективного використання пам'яті, оскільки компілятор може оптимізувати стек викликів. Інша форма рекурсії – взаємна (обопільна) рекурсія. Вона виникає тоді, коли дві або більше функцій викликають одна одну по черзі. Такий тип рекурсії може бути корисним у задачах, пов'язаних із пошуком шляхів, симуляцією складних систем.

Рекурсія є інструментом, який має як переваги, так і недоліки. Серед основних переваг можна виділити стислість і зрозумілість, що сприяє написанню компактного коду. До того ж рекурсія широко застосовується у функціональному програмуванні. Проте її використання має і свої недоліки. Для задач із занадто великою кількістю рівнів рекурсії може значно знижуватись продуктивність, порівняно з циклічними алгоритмами. Рекурсія в програмуванні не універсальна, оскільки певні мови програмування або середовища мають обмеження на її використання через проблеми з пам'яттю [2].

Метод рекурсії має у собі кілька небезпек. Найбільш очевидна небезпека – нескінченна рекурсія. У разі неправильної побудови алгоритму функція може пропускати або не досягати умови зупинки алгоритму і виконуватися нескінченно. Проблема полягає в тому, що нескінченність насправді означає роботу функції доти, поки не буде вичерпаний стековий простір. Такий алгоритм буде приводити до переповнення стека й аварійного завершення роботи програми [3].

Розглянемо реалізацію рекурсії у програмуванні. Більшість сучасних мов програмування підтримують рекурсію: Python, C#, JavaScript тощо. Але її ефективність буде залежати від правильного проєктування алгоритму. Рекурсивні алгоритми часто бувають доволі компактними, хоча можуть вимагати додаткових ресурсів пам'яті. Код на C#, що представлено нижче, демонструє рекурсивний алгоритм, який виконує обчислення факторіалу числа.

```
using System;
using System.Text;

0 references
class Recursion
{
    //Рекурсивна функція для обчислення факторіалу
    2 references
    static int Factorial(int n)
    {
        //Базовий випадок
        if (n == 0)
            return 1;

        //Рекурсивний випадок
        return n * Factorial(n - 1);
    }

    0 references
    static void Main(string[] args)
    {
        Console.OutputEncoding = Encoding.UTF8;

        Console.Write("Введіть число для обчислення факторіалу: ");
        int number = int.Parse(Console.ReadLine());

        if (number < 0)
        {
            Console.WriteLine("Факторіал визначається тільки для додатніх чисел!");
        }
        else
        {
            int result = Factorial(number);
            Console.WriteLine($"Факторіал числа {number} дорівнює {result}.");
        }
    }
}
```

Рисунок 1 – Код роботи рекурсивного алгоритму

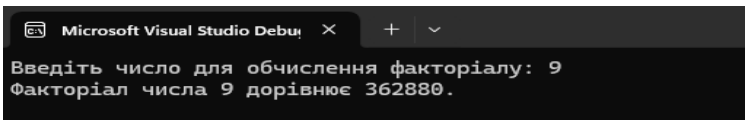


Рисунок 2 – Результат роботи коду

Список використаних джерел

1. Федорін І. В. Проектування та аналіз обчислювальних алгоритмів: Вступ до алгоритмів: навч. посіб. КПІ ім. Ігоря Сікорського, 2022. С. 19–24.

2. Foxminded. Рекурсія в програмуванні. URL: https://foxminded.ua/rekursiia-v-prohramuvanni/ (дата звернення 20.11.2024).

3. Основи інформатики та технологій програмування: навч. посіб. / М. Є. Рогоза, С. К. Рамазанов, А. В. Велігура, С. М. Танченко. Луганськ: Вид-во СНУ ім. В. Даля, 2012. С. 164–168.

УДК 004.9

Тищенко О. М., здобувач вищої освіти,
Поремський Ю. В., канд. техн. наук, старший викладач кафедри інформаційних технологій

АЛГОРИТМІЗАЦІЯ В ПОСТАНОВЦІ І РОЗВ'ЯЗАННІ ВІЙСЬКОВИХ ЗАВДАНЬ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритмізація є важливим аспектом сучасних військових технологій, які дають змогу розробляти ефективні стратегії та вирішувати складні проблеми, що виникають під час планування та виконання воєнних операцій. Він включає використання математичних моделей, спеціалізованих алгоритмів і комп'ютерних систем для оптимізації процесів управління, оцінки ризиків, ресурсного забезпечення та прийняття рішень [1].

В армії завдання варіюються від стратегії на полі бою до логістики постачання та обслуговування обладнання. Алгоритмізація дає змогу формулювати ці завдання в математичній або логічній формі, допомагаючи використовувати комп'ютерні програми для пошуку оптимальних рішень [2].

До основних категорій військових завдань, які потребують алгоритмізації, належать:

- Оптимальний розподіл ресурсів: наприклад, визначення оптимального розподілу боєприпасів, обладнання та персоналу.
- Планування операції: розробка ефективного плану нападу або захисту з урахуванням бойової ситуації, ресурсів і часу.
- Симуляція бою: розрахування результатів бою за допомогою математичних моделей, включно з мобілізацією військ, втратами, часом реакції та іншими факторами.

Використовуваний алгоритм може відрізнятися залежно від типу завдання. Ось кілька прикладів алгоритмів, які використовуються для вирішення військових завдань:

1. Алгоритм оптимізації:

- Алгоритм Дейкстри використовується для пошуку найкоротшого шляху між точками (наприклад, для мобільних пристроїв або для пошуку безпечного маршруту).
- Лінійне програмування для оптимізації розподілу ресурсів або логістики.

2. Моделі та моделювання:

- Теоретико-ігрова модель розрахунку стратегічних дій суперника в різних ситуаціях.
- Алгоритми обробки великих даних (big data), які можуть аналізувати інформацію з різних джерел для прогнозування дій противника та оптимізації власних дій.

3. Алгоритм розв'язання логістичних задач:

- Алгоритм Евкліда для розрахунку ефективного ресурсопостачання та маршрутів транспортування.
- Планувати алгоритми розгортання військ на опорних або передових позиціях з урахуванням можливостей противника та територіальних умов.

Алгоритми використовуються у всіх аспектах військової діяльності:

- Аналіз бойових можливостей: використання алгоритмів для оцінки ефективності бойових підрозділів і сил.
- Управління бойовими діями: визначення стратегічних і тактичних кроків, застосування різноманітних моделей для розрахунку ефективних дій у режимі реального часу.
- Система прийняття рішень: алгоритми для швидкого прийняття рішень на основі змін інформації (наприклад, команди під час бою).

Сучасні військові системи використовують складні програмні засоби для автоматизації вирішення проблем:

- Геоінформаційні системи (ГІС) забезпечують здатність оцінювати рельєф місцевості та планувати військові дії.
- Моделі прогнозування бойових дій, які використовують штучний інтелект і нейронні мережі для аналізу великої кількості змінних та даних.
- Інтегрована система команд у реальному часі, яка використовує різні алгоритми для коригування планів під час бою.

Алгоритми відіграють ключову роль у вирішенні військових завдань, оскільки дають змогу швидко та ефективно оцінювати ситуації, приймати найкращі рішення та забезпечувати правильне виконання дій. Застосування сучасних алгоритмів, особливо у сферах оптимізації, моделювання та управління ресурсами, необхідне для досягнення успіху у військових діях та забезпечення національної безпеки [3].

Список використаних джерел

1. Інформаційні технології у військовій справі. *UA5.ORG*. URL: https://ua5.org/technol/108-nformacjjn-tekhnolog-u-vjjskovjj-sprav.html (дата звернення: 30.11.2024).

2. Шарко О. Орієнтовний алгоритм дій щодо порядку прийняття на забезпечення у Збройних Силах України деяких видів засобів індивідуального захисту під час дії особливого періоду. URL: https://www.mil.gov.ua/content/ddz/TY_2024/%D0%90%D0%BB% D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%20%D0%97%D0%86%D0%97.pdf (дата звернення: 30.11.2024).

3. Олександр Зеленський. Алгоритм Дейкстри. URL: http://choippo.cn.sch.in.ua/Files/downloadcenter/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%20%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D0%B8.%20%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F.pdf (дата звернення: 30.11.2024).

УДК 004.94:519.6

Тітаренко Р. А., здобувач вищої освіти,
Комаров В. Ф., канд. техн. наук,
старший викладач кафедри прикладної
математики та кібербезпеки

ЗАСТОСУВАННЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ДЛЯ ОПТИМІЗАЦІЇ РОЗПОДІЛУ ЕЛЕКТРОЕНЕРГІЇ В МІСЬКИХ МЕРЕЖАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Математичне моделювання є потужним інструментом для оптимізації розподілу електроенергії у міських мережах. Використання математичних прогнозів та оптимізаційних алгоритмів дає змогу ефективно управляти енергетичними ресурсами, знижувати втрати електроенергії та підвищувати надійність мереж [1]. Це особливо важливо не тільки в умовах зростання населення та урбанізації, коли попит на електроенергію стрімко зростає, а й в умовах обмеженості енергетичних ресурсів під час російсько-української війни, коли агресор систематично намагається вивести з ладу всю енергетичну інфраструктуру нашої країни.

Існує кілька основних підходів до оптимізації розподілу електроенергії, зокрема:

1. Моделі оптимального потоку потужності (Optimal Power Flow – OPF): OPF моделі дають змогу визначити оптимальні режими роботи електричних мереж, враховуючи обмеження на генерування та споживання електроенергії [2].

2. Методи великих даних: використання аналізу великих даних допомагає виявляти закономірності у споживанні електроенергії та прогнозувати майбутні потреби [3].

3. Алгоритми керування втратами: алгоритми керування втратами спрямовані на зниження втрат електроенергії в мережі через оптимізацію маршрутів передачі та зменшення дисбалансу потужностей [4].

Наприклад, у дослідженні [2] показано, що використання моделей OPF допомагає знизити втрати енергії та підвищити стабільність роботи системи завдяки оптимізації параметрів роботи мережі та більш ефективному розподілу потужності. Різні методи для забезпечення оптимального розподілу електроенергії дали змогу підвищити стабільність і ефективність роботи електричних мереж, використання математичних моделей для оптимізації потоків потужності сприяло зниженню втрат енергії та поліпшенню загальної роботи системи.

Одним із успішних прикладів застосування аналізу даних для визначення ефективних алгоритмів оптимізації є дослідження [3], де аналіз великих даних

допоміг знизити втрати електроенергії на 15 % завдяки точному прогнозуванню попиту. У цьому дослідженні було використано дані з різних джерел для виявлення ключових факторів, що впливають на втрати, та розробку оптимальних стратегій їх зниження.

Дослідження [4] показало, що використання сучасних алгоритмів оптимізації дає змогу знизити втрати енергії на 10–20 % завдяки поліпшенню керування потоками електроенергії.

Оптимізація розподілу електроенергії за допомогою залучення математичного моделювання має значний потенціал для підвищення ефективності та надійності енергомереж. Подальші дослідження та впровадження сучасних оптимізаційних методів можуть забезпечити стійке та надійне енергопостачання в майбутньому, незважаючи на виклики часу. Впровадження подібних методів [1–4] у практику в нашій країні допоможе більш ефективно використовувати наявні ресурси та забезпечити стабільність електропостачання в умовах зростання попиту.

Список використаних джерел

1. Applications of optimization models for electricity distribution networks. URL: https://wires.onlinelibrary.wiley.com/doi/epdf/10.1002/wene.401 (дата звернення: 29.11.2024).

2. Optimal Power Flow in Distribution Network: A Review on Problem Formulation and Optimization Methods. URL: https://www.mdpi.com/1996-1073/16/16/5974 (дата звернення: 29.11.2024).

3. Optimization algorithm of power system line loss management using big data analytics. URL: https://energyinformatics.springeropen.com/articles/10.1186/s42162-024-00434-z (дата звернення: 30.11.2024).

4. Optimization Models in Electricity Markets. URL: https://assets.cambridge.org/97810094/16610/frontmatter/9781009416610_frontmatter.pdf (дата звернення: 30.11.2024).

УДК 004.422.5

Трохимчук О. М., здобувач вищої освіти,
Якубич К. О., асистент кафедри
інформаційних технологій

РОЛЬ АЛГОРИТМІЗАЦІЇ У РОЗВ'ЯЗАННІ ЗАДАЧ ШТУЧНОГО ІНТЕЛЕКТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритми та штучний інтелект (ШІ) є одними з найважливіших складників сучасних технологій. ШІ, як наука і практика, базується на розробці алгоритмів, здатних автоматично обробляти дані, приймати рішення і розв'язувати складні задачі. Алгоритми в ШІ – це чітко визначені послідовності дій, які забезпечують комп'ютеру здатність до навчання, пошуку оптимальних рішень та прогнозування. Завдяки розвитку ШІ ми можемо створювати інтелектуальні системи, що впроваджуються в різні галузі: від медицини до фінансів, від транспорту до освіти. В основі цих систем лежать різноманітні алгоритми: від базових методів пошуку та сортування до складних моделей машинного навчання та

глибоких нейронних мереж. У цьому матеріалі розглянуто ключові алгоритмічні концепції ШІ, зокрема жадібні, пошукові та навчальні методи, підкріплені практичними прикладами на мові програмування C#. Зазвичай алгоритм лінійного пошуку використовується в простих задачах пошуку, наприклад, у невпорядкованих даних [1].

Жадібні алгоритми (greedy algorithms) – це підхід у програмуванні та алгоритмах, коли на кожному кроці приймається рішення, яке здається найкращим у цей момент. Вони використовуються для розв'язання задач оптимізації, де мета – знайти оптимальне рішення, наприклад, мінімум чи максимум деякої функції. Основна ідея полягає в тому, що, вибираючи локально оптимальні кроки, можна досягти глобального оптимуму. Саме жадібні алгоритми часто використовуються у фінансових і оптимізаційних задачах [2]. Прикладом такого алгоритму є розрахунок решти монетами. Характеристики жадібних алгоритмів:

1. Жадібність на кожному кроці: алгоритм завжди вибирає найкраще доступне рішення на поточному етапі, не заглядаючи наперед.

2. Локальна оптимальність: кожне проміжне рішення є оптимальним з погляду локальної інформації.

3. Незмінність: рішення, прийняті на попередніх етапах, не змінюються на наступних.

Алгоритми пошуку є важливим інструментом у штучному інтелекті (ШІ), оскільки вони допомагають знаходити оптимальні шляхи, розв'язки задач або стратегії у складних просторах станів. Ці алгоритми застосовуються у задачах, де потрібно дослідити безліч можливих варіантів, як-от планування, ігри, маршрутизація, оптимізація тощо. Типи пошукових алгоритмів:

1. Неінформовані (або сліпі) пошуки. Ці алгоритми не мають інформації про цільову функцію, використовують лише структуру простору станів:

- Перебір у ширину (Breadth-First Search, BFS): досліджує всі вузли на поточному рівні перед переходом до наступного.
- Перебір у глибину (Depth-First Search, DFS): рухається по гілці графу до кінця, потім повертається назад.

2. Інформовані пошуки. Використовують евристичну інформацію для пришвидшення пошуку:

- Пошук «жадібний» (Greedy Search): вибирає вузол, який виглядає найкращим з погляду евристики.
- A* (A-star): об'єднує вартість до поточного вузла і оцінку вартості до цілі, що робить його оптимальним і повним алгоритмом.

3. Евристичні та оптимізаційні методи. Використовуються для задач з великими просторами станів:

- Генетичні алгоритми (Genetic Algorithms): імітують природний відбір для пошуку оптимальних рішень.
- Метод імітації відпалу (Simulated Annealing): імітує процес охолодження металу для знаходження глобального мінімуму.

4. Пошук у реальному часі. Використовується в системах, що працюють з динамічним середовищем:

- LRTA* (Learning Real-Time A-star): адаптується до змін під час виконання.

Пошук у ширину використовується в задачах на графах, наприклад, для навігації в соціальних мережах [3].

Нейронні мережі є основою сучасного штучного інтелекту (ШІ), даючи йому змогу обробляти великі обсяги даних і виявляти закономірності. Вони застосовуються в машинному навчанні для виконання завдань, як-от класифікація, прогнозування, переклад мов, розпізнавання мови та образів. Завдяки багаторівневій структурі (глибинне навчання) нейронні мережі можуть адаптуватися та навчатися з часом, підвищуючи точність і ефективність рішень [4]. Алгоритми є фундаментальною основою штучного інтелекту. Від простих жадібних методів до складних евристичних підходів і нейронних мереж, вони дають змогу створювати системи, які імітують мислення людини, автоматизують задачі та значно підвищують ефективність роботи в різних галузях.

Штучний інтелект уже зараз трансформує наше суспільство. Його можливості виходять далеко за межі теорії, забезпечуючи розв’язання реальних проблем – від прогнозування захворювань до управління трафіком у мегаполісах. Однак для ефективного застосування ШІ необхідно розуміти основні принципи алгоритмів, їх призначення та обмеження.

Знання алгоритмів і розуміння їх реалізації, як у представлених прикладах на C#, дає змогу не лише працювати з готовими рішеннями, а й створювати нові інтелектуальні системи. ШІ залишається відкритим полем для досліджень і вдосконалення, де правильне поєднання алгоритмів і креативності може привести до проривних інновацій.

Список використаних джерел

1. Жадібні алгоритми. URL: https://devzone.org.ua/post/zadibni-alhorytmy (дата звернення: 30.11.2024).

2. Алгоритми пошуку в ширину (BFS). URL: https://javarush.com/ua/quests/lectures/ua.javarush.python.core.lecture.level17.lecture07 (дата звернення: 30.11.2024).

3. unite.AI. URL: https://www.unite.ai/uk/%D0%B2%D0%B8%D1%81%D0% B2%D1%96%D1%82%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F (дата звернення: 30.11.2024).

УДК 004.001

Чайковський П. А., здобувач вищої освіти,
Штовба С. Д., д-р техн. наук, професор,
професор кафедри інформаційних технологій

ПОРІВНЯЛЬНИЙ АНАЛІЗ ТЕХНІК FEW-SHOT LEARNING ДЛЯ ДОМЕННОЇ АДАПТАЦІЇ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

Анотація. У статті проведено порівняльний аналіз сучасних технік few-shot learning для доменної адаптації великих мовних моделей (LLM). Розглянуто методи in-context learning, fine-tuning та meta-learning, їх переваги та обмеження в контексті адаптації до нових доменів з обмеженою кількістю даних.

Ключові слова: few-shot learning, доменна адаптація, великі мовні моделі, in-context learning, fine-tuning, meta-learning.

Великі мовні моделі (LLM) демонструють значні успіхи в обробці природної мови та широко використовуються в різних сферах [1]. Проте застосування цих моделей у нових доменах, де є обмежена кількість навчальних даних, часто призводить до зниження їх ефективності. У таких випадках важливим інструментом стають техніки few-shot learning, які сприяють ефективній адаптації моделей, забезпечуючи належний рівень продуктивності навіть за умов недостатньої кількості навчальної інформації.

Сучасні дослідження акцентують на критичній важливості здатності великих мовних моделей до адаптації в нових доменах з мінімальними витратами на підготовку даних [2]. Ефективність таких технік безпосередньо впливає на поширення LLM в різноманітних сферах, включно з медициною, фінансами, освітою та іншими [3]. Порівняння наявних підходів до few-shot learning, як-от in-context learning, fine-tuning та meta-learning, є необхідним для того, щоб зрозуміти їх потенціал і обмеження та обрати оптимальну стратегію в конкретних умовах.

Серед найбільш поширених технік few-shot learning для адаптації великих мовних моделей виокремлюються in-context learning, fine-tuning та meta-learning. Кожна з них має свої унікальні риси, які впливають на їх ефективність, вимоги до даних та ресурси, необхідні для реалізації.

In-Context Learning (ICL) базується на представленні моделі кількох прикладів у контексті запиту без оновлення ваг самої моделі [4]. Цей підхід дає змогу генерувати відповідь, враховуючи наданий контекст, що робить його надзвичайно корисним за умов обмежених даних. Однак, хоча ICL є найпростішим у реалізації і не потребує значних обчислювальних ресурсів, його точність у деяких випадках поступається іншим підходам, які передбачають навчання моделі.

Техніка fine-tuning, на відміну від ICL, передбачає адаптацію параметрів моделі на основі невеликої кількості даних із нового домену [5]. Цей підхід дає можливість моделі краще узгоджуватися зі специфікою даних, але вимагає обережності, оскільки існує ризик перенавчання. Застосування fine-tuning є доцільним, коли є достатня кількість прикладів для уникнення перенавчання та досягнення стабільних результатів.

Метод meta-learning орієнтований на швидку адаптацію до нових завдань шляхом попереднього навчання на різноманітних завданнях, що дає змогу моделі перенести знання в новий контекст з мінімальними витратами на навчання [6]. Цей підхід демонструє баланс між точністю та ефективністю, оскільки модель має можливість швидко адаптуватися до нових даних. Проте реалізація meta-learning є значно складнішою і потребує великого обсягу ресурсів на етапі попереднього навчання, що може обмежувати її застосування в умовах, де немає доступу до потужних обчислювальних потужностей.

Аналіз ефективності кожної з технік вказує на те, що вибір методу адаптації має залежати від особливостей завдання, обмежень у ресурсах та доступ-

ності навчальних даних. In-Context Learning доцільно застосовувати в ситуаціях, коли є необхідність у швидкій адаптації без додаткового навчання [3]. Fine-tuning може забезпечити більш високий рівень точності, особливо коли є достатній обсяг даних для налаштування моделі. Meta-learning, попри свої ресурсоємність та складність реалізації, забезпечує найкращі результати в умовах, коли модель повинна часто адаптуватися до нових завдань з мінімальними витратами на навчання.

Проведений аналіз дає змогу зробити висновок, що техніки few-shot learning є ефективним інструментом для доменної адаптації великих мовних моделей в умовах обмежених даних. Вибір конкретної техніки залежить від конкретних потреб і обмежень. У тих випадках, коли важлива швидка адаптація, рекомендується використання In-Context Learning, тоді як для досягнення високої точності доцільним є застосування fine-tuning. У випадках з частою зміною доменів і потребою в адаптації на основі мінімальних даних найкращим вибором є meta-learning.

Список використаних джерел

1. Language Models are Few-Shot Learners / T. B. Brown, B. Mann, N. Ryder Subbiah et al. URL: https://arxiv.org/abs/2005.14165 (дата звернення: 07.11.2024).

2. Evaluating Large Language Models Trained on Code / M. Chen, J. Tworek, H. Jun, Q. Yuan, P. de Oliveira et al. URL: https://arxiv.org/abs/2107.03374 (дата звернення: 07.11.2024).

3. Finn C., Abbeel P., Levine S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *Proceedings of the 34th International Conference on Machine Learning.* URL: https://arxiv.org/abs/1703.03400 (дата звернення: 07.11.2024).

4. Language Models are Unsupervised Multitask Learners / A. Radford., J. Wu., R. Child, D. Luan, D. Amodei, I. Sutskever. *OpenAI Blog.* URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (дата звернення: 07.11.2024).

5. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale / A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby. URL: https://arxiv.org/abs/2010.11929 (дата звернення: 07.11.2024).

6. Pre-trained Models for Natural Language Processing: A Survey / X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang. *Science China Technological Sciences.* 2021. № 63. P. 1872–1897. URL: https://arxiv.org/abs/2003.08271 (дата звернення: 07.11.2024).

УДК 519.872.5:519.21(043.2)

Чемес В. С., здобувач вищої освіти,
Сеник І. О., асистент кафедри
інформаційних технологій

АНАЛІЗ І ЗАСТОСУВАННЯ МОДЕЛЕЙ ТЕОРІЇ ЧЕРГ У СИСТЕМАХ ОБСЛУГОВУВАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах розвитку економіки та сектору послуг проблема ефективної організації систем обслуговування стає особливо важливою. Майже

щодня більшість людей стикаються з довгими чергами в магазинах, банках, лікарнях тощо. Неефективне обслуговування призводить до значних часових втрат клієнтів, зниження якості сервісу та економічних збитків для компанії.

Теорія черг – це математичний підхід до аналізу та оптимізації процесів обслуговування, коли потік клієнтів або об'єктів чекає на обслуговування на певних серверах або ресурсах. Для моделювання та дослідження різноманітних реальних ситуацій, як-от обслуговування клієнтів у банках, аеропортах, сервісних центрах або мережах передачі даних широко використовується теорія черг. За допомогою неї можна оптимізувати обслуговування клієнтів і витрати на утримання системи [1].

Основою теорії черг є декілька ключових понять, що описують поведінку клієнтів і системи загалом. Клієнт – це одиниця, яка потребує обслуговування і є потенційним учасником черги (наприклад, покупець у магазині або пакет даних у комп'ютерній мережі). Обслуговування надається спеціальними серверами, які, власне, і складають систему обслуговування, наприклад, касир у супермаркеті або сервер у дата-центрі. Потік клієнтів може бути випадковим або регулярним і визначається інтенсивністю, тобто кількістю клієнтів, які надходять у систему за одиницю часу (λ). Інтенсивність обслуговування відображає, скількох клієнтів сервер здатен обслужити за той самий проміжок часу (μ) [1, 2].

Системи черг класифікуються за типом потоку клієнтів, числом серверів і розподілом часу обслуговування. Наприклад, у позначенні M/M/1 кожна літера або цифра описує одну з характеристик системи.

Перша літера «M» (Markovian) означає, що потік клієнтів є випадковим і має розподіл Пуассона, що характеризує ситуації, коли клієнти надходять із певною середньою інтенсивністю, але з випадковими інтервалами між собою. Друга «M» вказує, що час обслуговування є експоненціально розподіленим, що характерно для процесів, у яких середня тривалість обслуговування постійна, але кожен окремий випадок може значно відхилятися. Остання цифра вказує кількість серверів: у моделі M/M/1 – один сервер, який обслуговує клієнтів послідовно, тому черги можуть бути довшими за високої інтенсивності потоку.

У моделі M/M/s структура подібна, проте «s» позначає наявність кількох серверів, які працюють одночасно. Це дає змогу обслуговувати кількох клієнтів паралельно, знижуючи час очікування і підвищуючи пропускну здатність системи. Для моделі M/M/s, де є кілька серверів, основні формули для розрахунку характеристик системи змінюються і враховують кількість серверів [3, 4].

Середня кількість клієнтів у системі відображає середнє число клієнтів, які одночасно перебувають у системі, включно з тими, які чекають у черзі та обслуговуються:

$$L = \frac{\lambda}{\mu} \cdot \left[\frac{1}{(1-p)}\right],$$

де коефіцієнт завантаження кожного сервера:

$$p = \frac{\lambda}{\mu \cdot s}.$$

Середній час перебування клієнта в системі вказує на час, який клієнт проводить у системі від моменту прибуття до завершення обслуговування:

$$W = \frac{1}{\mu} \cdot \left[\frac{1}{(1-p)}\right].$$

Середня кількість клієнтів у черзі – це кількість клієнтів, які очікують на обслуговування:

$$L_q = \frac{\lambda^2}{\mu \cdot (s \cdot (\mu - \lambda))} \cdot \left[\frac{1}{(1-p)}\right].$$

Середній час очікування в черзі відображає час, який клієнт проводить у черзі перед обслуговуванням:

$$W_q = \frac{\lambda}{\mu \cdot s \cdot (\mu - \lambda)} \cdot \left[\frac{1}{(1-p)}\right] \text{ [4].}$$

Розглянемо приклад:

У великому супермаркеті є 4 каси, що працюють за системою M/M/s. Відомо, що середньоденна кількість покупців у магазині становить 480 клієнтів, а кожен касир обслуговує в середньому 20 клієнтів за годину. Каси працюють 8 годин на день. Оскільки наявна система часто призводить до довгих черг, керівництво хоче оптимізувати кількість кас, щоб зменшити час очікування клієнтів і покращити ефективність обслуговування.

Задача полягає в тому, щоб визначити оптимальну кількість кас за умов:

- Витрати на одного касира – 15 000 грн/місяць.
- Вартість втрат від очікування клієнтів – 100 грн/година.
- Бюджет на оптимізацію – 110 000 грн/місяць.

Розв’язання:

Потік клієнтів λ визначається як середня кількість клієнтів, які прибувають до системи за одиницю часу. Якщо в день магазин обслуговує 480 клієнтів за 8 годин, то потік клієнтів на годину буде:

$$\lambda = \frac{480}{8} = 60 \text{ клієнтів за годину.}$$

Кожен касир обслуговує 20 клієнтів на годину. Тому інтенсивність обслуговування на кожній касі:

$$\mu = 20 \text{ клієнтів на годину.}$$

Якщо для кожного варіанта ми обираємо кількість кас s, то загальні витрати на зарплату касирів за місяць будуть:

$$Z_{\text{зарплата}} = s \cdot 15\,000 \text{ грн/місяць.}$$

Загальні витрати на втрати через очікування:

$$Z_{\text{втрати}} = L_q \cdot W_q \cdot 100 \text{ грн/год.}$$

Для кожного варіанта кількості кас ссс будемо обчислювати загальні витрати Z_{total}, що включають витрати на зарплату касирів та витрати від втрат через очікування:

$$Z_{total} = Z_{\text{зарплата}} + Z_{\text{втрати}};$$

Нам потрібно вибрати таку кількість кас s, яка мінімізує Z_{total}, але водночас не перевищує бюджет у 110 000 грн/місяць.

Виконавши розрахунки, отримаємо таблицю з результатами.

Таблиця 1 – Результати обчислень

Кількість кас	**Інтенсивність завантаження**	**Середній час очікування в черзі**	**Середня кількість клієнтів у черзі**	**Витрати на зарплату**	**Втрати через очікування**	**Загальні витрати**
4	0,75	9	45	60 000	40 500	100 500
5	0,6	6	36	75 000	21 600	96 600
6	0,5	4,5	24	90 000	10 800	100 800
7	0,43	3,2	22,4	105 000	5 600	110 600

В результаті, для бюджету 110 000 грн/місяць оптимальним вибором буде 6 кас, оскільки загальні витрати становлять 100,800 грн/місяць, що є менше за бюджет.

Висновки. Отже, можна моделювати реальні процеси, аналізувати їх особливості та оптимізувати роботу систем за допомогою методів теорії черг. Параметри потоку клієнтів, рівень завантаженості серверів і час обслуговування можна точно оцінити за допомогою моделей M/M/1 і M/M/s. Використання теорії черг є важливою частиною процесу вдосконалення систем обслуговування, що сприяє розвитку сфери послуг і гарантує високий рівень задоволеності клієнтів.

Список використаних джерел

1. KA Group. URL: https://kagroup.ua/tpost/iodsc61fj1-teorya-cherg (дата звернення: 25.11.2024).

2. Wikipedia. URL: https://en.wikipedia.org/wiki/Queueing_theory (дата звернення: 25.11.2024).

3. Science Direct. URL: https://www.sciencedirect.com/topics/computer-science/queueing-theory (дата звернення: 25.11.2024).

4. Real Statistics Using Excel. URL: https://real-statistics.com/probability-functions/queueing-theory/m-m-s-queueing-model/ (дата звернення: 25.11.2024).

УДК 004.422.5

Чулюк В. В., здобувач вищої освіти,
Якубич К. А., асистент кафедри
інформаційних технологій

СУТНІСТЬ ТА ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МЕТОДУ СОРТУВАННЯ ВСТАВКОЮ

Донецький національний університет імені Василя Стуса, м. Вінниця

Для структуризації та впорядкування різноманітних масивів даних і для їх подання у певному вигляді використовуються алгоритми сортування. Для сортування даних було розроблено безліч алгоритмів, що мають як свої переваги, так і недоліки. З розвитком розподілених систем та паралельних обчислень розвиваються і алгоритми з використанням цих технологій. Такий підхід вдосконалює та підвищує ефективність самих алгоритмів з використанням паралельних технологій. Виникає необхідність порівняння ефективності алгоритмів, що використовують технології розподілених систем та паралельних обчислень. Також можливість порівняння ефективності алгоритмів є цікавою та пізнавальною в навчальному процесі підготовки ІТ-спеціалістів [1].

Відомо багато методів сортування масиву, що відрізняються швидкодією й обсягом оперативної пам’яті, яка під час цього використовується. Серед цих методів можна виділити методи внутрішнього та зовнішнього сортування.

Методи внутрішнього сортування не передбачають використання допоміжних масивів. Ці методи застосовують до масивів, що повністю розташовані в оперативній пам’яті.

Методи зовнішнього сортування застосовують до великих масивів даних, які зберігаються на зовнішніх носіях. Методи внутрішнього сортування прийнято поділяти на дві групи: елементарні (прямі) та удосконалені методи [2]. Найбільш відомий елементарний метод сортування масиву – сортування вставкою (включенням).

Цей метод полягає у тому, що кожен елемент послідовно вставляється у вже відсортований список. Розмір уже відсортованого списку спочатку дорівнює одиниці. Алгоритм сортування вставкою забезпечує сортування перших k елементів після k-ї ітерації. У результаті відсортована частина збільшується на один елемент, а невідсортована – на один елемент зменшується [3].

Отже, на кожному кроці алгоритму сортування методом вставки треба виконати дві операції:

- пошук позиції для вставки елементу в лівій (відсортованій) частині;
- власне його вставку з подальшим зсувом на одну позицію вправо від елементів відсортованої частини.

Програма нижче моделює роботу сортування масиву методом вставки та виводить задіяну кількість операцій, використовуючи мову програмування C#, код демонструє структуру цього методу, реалізованого за допомогою циклів «for» та «while».

```
using System;

class Program
{
// Алгоритм сортування вставками (Insertion Sort)
    1 usage
    static void InsertionSort(int[] arr)
    {
        int n = arr.Length;
        int operations = 0; // Лічильник операцій

        for (int i = 1; i < n; i++) // Починаємо з другого елемента
        {
            int key = arr[i]; // Поточний елемент, який потрібно вставити
            int j = i - 1;

            // Зсуваємо елементи вправо, щоб знайти правильне місце для вставки
            while (j >= 0 && arr[j] > key)
            {
                operations++; // Кожне порівняння додається до кількості операцій
                arr[j + 1] = arr[j];
                j--;
            }

            arr[j + 1] = key; // Вставляємо елемент у правильну позицію
            operations++;
        }

        // Виводимо результат
        Console.WriteLine($"Метод сортування вставкою: {string.Join(", ", arr)}");
        Console.WriteLine($"Кількість операцій: {operations}\n");
    }
```

Рисунок 1 – Код роботи сортування масиву методом вставки

```
// Головна функція
    static void Main()
    {
        int[] array = { 50, 80, 90, 40, 20, 70, 55, 10, 85, 30 };

        Console.WriteLine("Початковий масив: " + string.Join(", ", array) + "\n");

        // Метод сортування вставкою
        int[] insertionArray = (int[])array.Clone();
        InsertionSort(insertionArray);
    }
}
```

Рисунок 2 – Продовження коду

```
Початковий масив: 50, 80, 90, 40, 20, 70, 55, 10, 85, 30

В?дсортований масив методом вставки: 10, 20, 30, 40, 50, 55, 70, 80, 85, 90
К-кть операц?й: 36
```

Рисунок 3 – Результат коду

Цей код демонструє алгоритм сортування вставкою у мові програмування C#. Він ілюструє, як елементи по черзі вставляються у правильне місце від-

сортованої частини масиву. Алгоритм підходить для невеликих масивів через свою простоту, але для великих масивів він може працювати повільно через високу кількість операцій у випадку невпорядкованих даних. Сортування методом вставки відіграє фундаментальну роль у багатьох сферах програмування та інформаційних технологій. Це базовий, надійний і простий у реалізації алгоритм, який знаходить застосування в широкому спектрі задач – від навчальних прикладів до складних систем обробки даних.

Список використаних джерел

1. Програмна реалізація та дослідження алгоритмів паралельного швидкого сортування / В. О. Денисюк , Н. А. Потапова, О. В. Зелінська, М. Б. Тарасюк. *Вісник Хмельницького національного університету. Технічні науки.* 2023. № 4. С. 95–105. URL: http://journals.khnu.km.ua/vestnik/?page_id=41 (дата звернення: 05.12.2024).

2. Ковалюк Т. В. Основи програмування: підручник. Київ: Вид-во BHV, 2005. 384 с.

3. GURU99. URL: https://www.guru99.com/uk/insertion-sort-algorithm.html (дата звернення 05.12.2024).

УДК 004.42

Щербина Д. С., здобувач вищої освіти,
Потапова Н. А., канд. екон. наук., доцент,
доцент кафедри інформаційних технологій

ЗАСТОСУВАННЯ АЛГОРИТМУ АЛЬФА-БЕТА ВІДСІКАННЯ ДЛЯ ОПТИМІЗАЦІЇ ШАХОВИХ РІШЕНЬ

Донецький національний університет імені Василя Стуса, м. Вінниця

З кожним роком шахи стають більш популярними. Вони виросли з простої гри у великий спорт, і сприяють розвитку мислення, оскільки поєднують стратегічне і тактичне планування з логічним аналізом. Хоча успіх у спорті прийшов не відразу, проте за шахами вже багато років слідкують галузі алгоритмічного програмування та штучного інтелекту. Цей інтерес сприяє розвитку шахових програм, які постійно вдосконалюються для того, щоб вказувати як на помилки досвідчених шахістів, так і для навчання новачків.

Однією з ключових задач, які насамперед мають виконувати різні шахові програми, є забезпечення швидкої і точної оцінки позиції, щоб обрати оптимальний хід або варіант. Одним з перших алгоритмів, який застосовувався для знаходження оптимального рішення, був мінімакс. Але він не є ефективним, оскільки потрібно перевірити всі варіанти ходів, що, особливо в шахах, є дуже ресурсомістким процесом. Тому щоб оптимізувати цей алгоритм, було розроблено алгоритм альфа-бета відсікання. Цей метод, обрізаючи непотрібні гілки дерева рішень, дає змогу скоротити час та ресурси на пошук оптимального рішення [1].

Алгоритм альфа-бета відсікання відрізняється від мінімаксу тим, що він використовує не повний перебір усіх варіантів, а додає обмеження, які нази-

ваються відповідно до самого алгоритму – альфа та бета. У цьому алгоритмі альфа – це мінімальне або максимальне значення, яке буде задовольняти поставлену задачу, а бета – це ті значення, які ми отримаємо під час перегляду того чи іншого рішення [2]. Звідси випливає те, що поки значення бета перевершує альфа, то гілка рішень розглядається, а коли бета вже ніяк не зможе бути кращим за альфа, то гілка рішень відсікається. Наприклад, коли під час шахової партії гравцю загрожують взяттям ферзя (королеви), то немає сенсу обирати варіанти наступних ходів методом повного перебору, оскільки гравець втратить ферзя після першого ж. Тому в цій ситуації на допомогу приходить альфа-бета відсікання, де альфа – оцінка позиції, а бета – оцінка після пари ходів (білі-чорні), і оскільки не всі ходи рятують ферзя, що значно погіршить оцінку позиції – бета, тому такі варіанти відсікаються і далі їх обрахунок не ведеться.

Використовуючи алгоритм альфа-бета відсікання, можна отримати низку переваг, що робить його одним з основних методів, які використовуються під час створення шахових програм. Це такі переваги:

- зменшення можливих варіантів шляхом відсікання завідомо гірших, дає змогу пришвидшити аналіз позицій, що допоможе зберегти ресурси та час, який потрібно витратити на знаходження оптимального ходу;
- оскільки не потрібно витрачати ресурси на обрахунок непотрібних варіантів, то за допомогою цього можна підвищити глибину розрахунків, що збільшить точність рішення;
- можна реалізовувати шахові програми на мобільних пристроях. Оскільки знижуються вимоги до ресурсів, які потрібно витрачати на аналіз позицій, то більш слабкі пристрої можуть самостійно виконувати такі операції.

Хоча алгоритм альфа-бета відсікання має плюси щодо швидкості та економії ресурсів, але бувають випадки, коли відбувається «позиційна» жертва. Тобто ми можемо віддати матеріал заради отримання ініціативи для того, щоб у подальшому конвертувати цю ініціативу в матеріальну перевагу. Такі речі важко розпізнати тільки розрахунками, тому під час створення шахових програм із цим алгоритмом потрібно враховувати різні особливості, які притаманні шахам. Такими особливостями можуть бути шахова геометрія (зв’язки, подвійні удари, рентгени тощо) або розташування інших фігур на дошці, які можуть непрямо впливати на прийняття того чи іншого рішення. Врахування всіх додаткових факторів дасть змогу максимально чітко робити аналіз позицій [3].

Застосування алгоритму альфа-бета відсікання в шахах може бути дуже різноманітним. Окрім використання для стандартного аналізу позиції, його можна використовувати для гри проти людини або для вирішення різних задач. Цей алгоритм можна налаштувати так, щоб він підлаштовувався під відповідний рівень гравця, наприклад, для гри з початківцем аналізувати позицію на 1 або 2 ходи вперед, а зі зростанням рівня гравця – збільшувати глибину розрахунків. Схожий напрям можна використати для розв’язання задач, коли для різного рівня гравців будуть створюватися задачі різної складності, проте з однаковими умовами їх виконання.

Підсумовуючи можна сказати, що альфа-бета відсікання є ключовим алгоритмом під час створення шахових програм. Він зможе забезпечити швидке

знаходження оптимального варіанта, адаптивний підхід до користувачів залежно від їх рівня, а у разі його розширення чи комбінації з іншими методами можна створити повноцінний шаховий двигун, який зможе давати точну оцінку навіть на достатній глибині розрахунків.

Список використаних джерел

1. Fuller S. H., Gaschnig J. G., Gillogly J. J. An analysis of the alpha–beta pruning algorithm. Department of Computer Science Report, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1973, 51 p.

2. Felstiner C. Alpha-Beta Pruning, Whitman College, 2019.

3. Wang J. J., Liu M. S., Zhao G. D. Design of Military Chess System Based on Alpha-Beta Pruning Algorithm. *36th Chinese Control and Decision Conference (CCDC).* IEEE, 2024. P. 3092–3097.

УДК 004.08

Рудь О. С., здобувачка вищої освіти,
Юстименко Є. А., здобувач вищої освіти,
Ніколюк П. К., д-р фіз.-мат. наук, професор
кафедри інформаційних технологій

АЛГОРИТМИ ШИФРУВАННЯ ДЛЯ ЗАХИСТУ РАДІОЗВ'ЯЗКУ В УМОВАХ БОЙОВИХ ДІЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні воєнні операції все більше залежать від швидкої та безпечної передачі інформації. Радіозв'язок є надважливим засобом комунікації на полі бою, проте він залишається вразливим до перехоплення та глушіння з боку супротивника. Розробка ефективних алгоритмів шифрування має велике значення для забезпечення конфіденційності та достовірності переданих даних.

Ця тема є надзвичайно актуальною в умовах воєнного стану і в час, коли технологічні засоби ведення бойових дій постійно вдосконалюються. Під час бойових дій інформаційна безпека особливо важлива та безпосередньо впливає на успішне виконання військових завдань. Забезпечення захищеної передачі даних допомагає запобігти перехопленню ворогом стратегічно важливої інформації та зберегти життя військовослужбовців.

Радіозв'язок є одним із ключових елементів забезпечення військових операцій, адже через нього передаються накази, координати, розвідувальні дані та інша критично важлива інформація. Одним із найпоширеніших методів захисту даних є шифрування. Проте розробка алгоритмів шифрування для військових потреб має враховувати специфічні вимоги до швидкості, безпеки та енергоефективності [1].

Процес безпечної передачі даних включає кілька етапів: шифрування даних на стороні передавача, передача зашифрованих даних через радіоканал і розшифрування даних на стороні приймача. На рисунку 1 зображено етапи безпечної передачі даних між передавачем і приймачем у радіозв'язку.

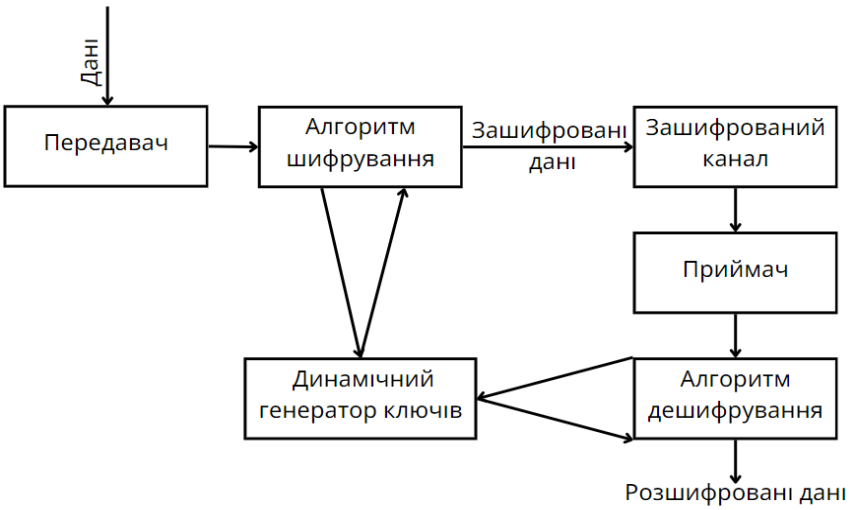


Рисунок 1 – Процес безпечної передачі даних між передавачем і приймачем у радіозв'язку

Опис зв'язків між елементами схеми:

1. Передавач:

- Передавач отримує дані, які потрібно передати, і надсилає їх у блок алгоритму шифрування.
- Для виконання шифрування цей блок взаємодіє з динамічним генератором ключів, який генерує унікальний ключ для кожного повідомлення чи пакету даних.
- Зашифровані дані надсилаються через зашифрований канал до приймача.

2. Зашифрований канал:

- Канал передачі (радіосигнал) слугує середовищем для пересилання зашифрованих даних від передавача до приймача.

3. Приймач:

- Приймач отримує зашифровані дані через зашифрований канал.
- Зашифровані дані передаються до блоку алгоритму дешифрування, який відновлює вихідні дані.
- Блок дешифрування також використовує динамічний генератор ключів, щоб створити той самий ключ, що й на передавачі, синхронізований за допомогою попередньо узгодженого секрету або параметрів радіоканалу.

4. Динамічний генератор ключів:

- У передавачі й приймачі генератори працюють синхронно, створюючи однакові ключі в реальному часі. Це дає змогу уникнути передачі ключів через канал зв'язку, що знижує ризик їх перехоплення [2].

Перехоплення радіосигналу є звичною практикою на полі бою. Тому алгоритми шифрування повинні бути максимально стійкими до криптоаналізу, навіть якщо противник має доступ до значних обчислювальних ресурсів. Класичні алгоритми, як-от AES (Advanced Encryption Standard), хоча і забезпечують високий рівень безпеки, часто виявляються непридатними для використання у військових пристроях, як-от портативні радіостанції чи дрони. Ці пристрої мають обмежену обчислювальну потужність, що робить ресурсомісткі алгоритми неприйнятними.

Легковагові шифри, наприклад, SPECK або SIMON, є перспективними для таких умов. Вони спеціально розроблені для роботи в обмежених апаратних середовищах і мають оптимізовану структуру, що знижує енерговитрати. Ці шифри використовують прості математичні операції, які виконуються швидко навіть на пристроях із низькою продуктивністю, водночас забезпечуючи захист від базових атак.

Ще одним важливим елементом у військовому зв'язку є динамічне управління ключами. Традиційні підходи до генерації і зберігання ключів не враховують специфіку бойових умов, де радіосигнал може бути перехоплений або заглушений. Для уникнення таких ризиків сучасні алгоритми пропонують генерувати ключі у реальному часі. Наприклад, можна використовувати псевдовипадкові генератори, які створюють унікальний ключ для кожного пакету даних. Початковий секрет, відомий лише сторонам комунікації, дає змогу синхронізувати генерацію ключів без їх передачі через радіоканал.

Особливої уваги заслуговують інноваційні методи генерації ключів, що базуються на фізичних параметрах середовища, як-от шум радіоканалу, фазові зміщення чи затримка сигналу. Такі параметри є унікальними для кожної ситуації, що ускладнює їх прогнозування або підробку супротивником.

Для підвищення загальної безпеки ефективним є використання гібридних підходів, які поєднують симетричне і асиметричне шифрування. У таких системах асиметричне шифрування використовується для захищеної передачі ключів, а симетричне – для шифрування великих обсягів даних. Така комбінація дає змогу досягти високої швидкості передачі за збереження надійності захисту.

Енергоефективність також є критичним параметром для алгоритмів шифрування. Військові пристрої часто працюють на батарейках чи інших автономних джерелах енергії, і їх ресурс може бути обмеженим. Оптимізація алгоритмів із використанням менш енерговитратних операцій, як-от додавання чи побітові зсуви замість множення або експоненціювання, дає змогу продовжити час роботи пристрою без заміни або заряджання батареї.

Важливо також враховувати стійкість алгоритмів до атак «людина посередині» (MITM) та глушіння сигналу. Використання технік, як-от зміна частоти передачі та додаткове шифрування контрольних пакетів, ускладнює роботу супротивника і підвищує загальну надійність системи

Захист радіозв'язку в умовах бойових дій є не лише технічною, але й стратегічною проблемою, адже ворог завжди намагається використовувати інформаційні слабкості для отримання переваги. Також важливо враховувати необхідність інтеграції розроблених алгоритмів у вже наявну військову інфраструктуру без значних витрат на її модернізацію. Використання динамічних шифрів з підтримкою адаптації до змін умов середовища дасть змогу мінімізувати ризики перехоплення або глушіння сигналу. До того ж приділяти увагу розвитку систем резервного зв'язку, які автоматично перемикаються на альтернативні канали в разі атак. Надійне шифрування відіграє значну роль не лише у захисті інформації, а й у збереженні життя військових, адже перехоплення даних може розкрити позиції підрозділів. Подальші дослідження у цій сфері мають зосередитися на спрощенні апаратної реалізації складних алгоритмів, щоб вони стали доступними навіть для малих мобільних пристроїв [3].

Отже, для захисту радіозв'язку у бойових умовах необхідно враховувати широкий спектр факторів, від апаратних обмежень до криптографічних загроз. Легковагові шифри, динамічне управління ключами та адаптивність до параметрів середовища є основними компонентами сучасних алгоритмів, які забезпечують швидку, надійну та захищену передачу інформації.

Список використаних джерел

1. Шолудько В. Г., Єсаулов М. Ю. Організація військового зв'язку: навч. посіб. Київ: 2017. 112 с.
2. Кутень Р. Б., Синявський О. Ю. Методи і засоби забезпечення стабільності та захисту радіозв'язку в умовах складної електромагнітної обстановки: наукова стаття. Львів: 2024.
3. Головін Ю. О. Основи радіозв'язку з рухомими об'єктами: навч. посіб. Київ: 2016. 14 с.

УДК 004.75:004.021

Ярошенко Б. М., здобувач вищої освіти,
Хмелівський Ю. С., асистент кафедри
інформаційних технологій

РОЛЬ АЛГОРИТМІВ У БЛОКЧЕЙН-ТЕХНОЛОГІЯХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Блокчейн-технології є ключовим елементом сучасної цифрової економіки, забезпечуючи децентралізацію, прозорість і безпеку. Ці технології стали основою для інновацій, як-от криптовалюти, смарт-контракти та децентралізовані додатки (DApps). Алгоритми відіграють вирішальну роль у функціонуванні блокчейнів, забезпечуючи узгодженість даних, безпеку і масштабованість системи. Основні типи алгоритмів у блокчейні можна розділити на такі категорії:

1. Алгоритми консенсусу – це механізм, за допомогою якого блокчейн досягає узгодженості даних серед усіх учасників мережі без необхідності централізованих регуляторів. Оскільки мережа є одноранговою, процес перевірки транзакцій має бути автоматизованим. Алгоритми консенсусу забезпечують, щоб усі учасники мережі дотримувалися правил і здійснювали транзакції в належний спосіб. Вони також запобігають спробам подвійного витрачання цифрових валют, що є важливим для забезпечення безпеки. Серед найбільш відомих алгоритмів консенсусу – Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS) [1].

2. Алгоритми хешування – це математична операція, яка приймає вхідні дані будь-якого розміру, обробляє їх і створює вихідні дані фіксованого розміру, відомі як хеш. У блокчейн-технологіях основними функціями цього алгоритму є підвищення безпеки, оскільки будь-яка зміна даних змінює хеш блоку, роблячи втручання очевидним. Це забезпечує цілісність даних, оскільки наступний блок містить хеш попереднього, і зміна даних у попередньому блоці робить ланцюг недійсним. Хешування також допомагає перевіряти дані – порівнюючи хеш блоку з хешем, на який посилається наступний блок, можна підтвердити, що дані не були змінені [2].

3. Структури даних є важливою частиною ефективної роботи блокчейн-технологій, оскільки правильна організація даних впливає на швидкість і надійність системи. Однією з основних структур є дерево Меркля (Merkle Tree), яке використовується для ефективної перевірки транзакцій у блоці. Завдяки цій структурі зменшується обсяг інформації, яку необхідно зберігати, що забезпечує високу швидкість перевірки. Дерево Меркля дає змогу перевіряти транзакції з мінімальними витратами ресурсів завдяки зменшенню кількості необхідних для перевірки даних. Також важливими є зв'язаний список (Linked List), який можна використовувати для представлення блоків у блокчейн-системі, де кожен блок містить хеш попереднього блоку, що утворює ланцюг [3].

4. Алгоритми пошуку та сортування сприяють ефективному управлінню транзакціями і доступу до даних. Наприклад, сортування транзакцій за розміром комісії дає змогу майнерам обробляти транзакції з найвищим пріоритетом, що оптимізує використання мережевих ресурсів. Для пошуку та доступу до даних використовуються структури, як-от хеш-таблиці, які забезпечують швидкий доступ до таких даних, наприклад, транзакції або баланси рахунків [4].

Алгоритми є основою ефективного функціонування блокчейн-технологій, забезпечуючи безпеку, децентралізацію та прозорість мереж. Основними типами алгоритмів є алгоритми консенсусу (PoW, PoS, PBFT тощо), хешування, а також структури даних, як-от дерево Меркля, що дають змогу ефективно перевіряти транзакції та зберігати цілісність даних. Правильне розуміння і застосування цих алгоритмів критичне для забезпечення стабільності та безпеки блокчейн-систем.

Список використаних джерел

1. Exbase. Алгоритм консенсусу. URL: https://exbase.io/uk/wiki/algoritm-konsensusu (дата звернення 02.12.2024).

2. Plisio. Як працює хешування в блокчейні? 2024. URL: https://plisio.net/uk/blog/how-does-hashing-in-blockchain-work (дата звернення 02.12.2024).

3. Merkle Tree: A Fundamental Component of Blockchains / H. Liu, X. Luo, H. Liu, X. Xia. 2021. URL: https://ieeexplore.ieee.org/document/9588047 (дата звернення 02.12.2024).

УДК 004.414.23:641.5

Кузьміна М. О., здобувачка вищої освіти,
Якубич К. О., асистент кафедри
інформаційних технологій

РОЗРОБКА ПРОГРАМИ ДЛЯ КУХАРІВ НА PYTHON

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Сучасна кухня все частіше стає місцем впровадження новітніх технологій. Програми для кухарів дають змогу зручно управляти рецептами, формувати списки продуктів, планувати раціон та оптимізувати час. Python, як одна з найпопулярніших мов програмування, пропонує широкий спектр інструментів для створення інтуїтивних та ефективних рішень у цій сфері.

Завдяки простоті синтаксису та багатим бібліотекам ця мова стає вибором номер один для розробників, які прагнуть створювати корисні додатки [1].

Метою цієї роботи є створення програми для кухарів, яка не лише автоматизує рутинні завдання, але й допомагає професіоналам та аматорам готувати з легкістю та задоволенням.

Актуальність цієї теми зумовлена зростанням попиту на програми, які допомагають вирішувати щоденні завдання, як-от розрахунок харчової цінності страв, підбір рецептів відповідно до наявних продуктів та генерація списків покупок. До того ж цифрові інструменти для кухарів важливі для впровадження принципів здорового харчування, оскільки вони дають змогу враховувати калорійність та інші параметри.

Функції програми. Програма для кухарів на Python розроблена так, щоб виконувати широкий спектр завдань.

Програма дає можливість зручно управляти базою рецептів. Користувачі можуть додавати нові рецепти або редагувати вже наявні, зберігаючи детальну інформацію про інгредієнти, етапи приготування та час, необхідний для готування.

Особливістю програми є вбудований калькулятор калорійності та харчової цінності. Він автоматично розраховує калорії, а також кількість білків, жирів і вуглеводів у кожній страві на основі її інгредієнтів. Користувач може масштабувати рецепти залежно від бажаної кількості порцій, і програма автоматично адаптує всі обчислення.

Для полегшення планування покупок програма може створювати списки продуктів на основі вибраних рецептів. Ці списки можна експортувати у текстовий формат або в додатки для покупок, що значно спрощує організацію приготування страв.

Це універсальне рішення для професійних кухарів і кулінарних ентузіастів, яке значно полегшує процес приготування їжі, планування раціону та управління запасами продуктів.

У лістингу 1 наведений приклад коду, який розраховує калорійність страви.

Лістинг 1

```
import pandas as pd
# Дані про інгредієнти
data = {
   ‘ingredient’: [‘Молоко’, ‘Цукор’, ‘Борошно’],
   ‘calories’: [42, 387, 364],
   ‘weight’: [100, 100, 100]
}
# Функція для розрахунку калорійності
def calculate_calories(recipe):
   total_calories = 0
   for item in recipe:
      for index, row in data.iterrows():
         if row[‘ingredient’] == item[‘name’]:
            calories_per_unit = row[‘calories’] / row[‘weight’]
            total_calories += calories_per_unit * item[‘quantity’]
   return total_calories
# Приклад страви
```

```
recipe = [{‘name’: ‘Молоко’, ‘quantity’: 200}, {‘name’: ‘Цукор’, ‘quantity’: 50}]
print(f"Калорійність страви: {calculate_calories(recipe)} ккал")
```

Результат програми: *Калорійність страви: 277.5 ккал*

Висновки. Розробка програми для кухарів на Python демонструє значний потенціал для автоматизації кулінарних процесів. Використання бібліотек pandas, matplotlib та інших дає змогу створити функціональний інструмент, що задовольнить потреби як професійних кухарів, так і любителів. Програма зменшує час, витрачений на планування страв, сприяє дотриманню здорового харчування і полегшує управління ресурсами на кухні [4].

Список використаних джерел

1. Lutz M. Learning Python. O’Reilly Media. 2013. 1643 p.
2. VanderPlas J. Python Data Science Handbook. O’Reilly Media, 2016. 548 p.
3. Taibi D., Lenarduzzi V., Pahl C. Processes, Motivations, and Issues for Migrating to Microservices Architectures. *IEEE Software*. 2017. P. 22–32.
4. Документація Python. URL: https://docs.python.org. (дата звернення: 01.12.2024).

УДК 004.415:004.432.4

Ілик В. В., здобувач вищої освіти,
Сеник І. О., асистент кафедри
інформаційних технологій

АРХІТЕКТУРА МІКРОСЕРВІСІВ ДЛЯ МАСШТАБНИХ КЛІЄНТ-СЕРВЕРНИХ СИСТЕМ

Донецький національний університет імені Василя Стуса, м. Вінниця

В епоху цифрової трансформації компанії потребують архітектурних рішень, які забезпечують швидке реагування на зміни, гнучкість у розвитку та високу масштабованість. Мікросервісна архітектура стає все популярнішою через можливість адаптації до змінного навантаження та здатність задовольняти вимоги високонавантажених систем.

Останнім часом мікросервіси почали широко застосовуватися у великих компаніях (Netflix, Amazon, Uber) для розробки високонадійних, продуктивних систем.

Порівняння з традиційним монолітним підходом, де додаток розробляється як єдине ціле, ускладнює гнучкість та масштабування.

Мета роботи – проаналізувати основні переваги та виклики впровадження мікросервісної архітектури у клієнт-серверні системи, виявити умови, за яких вона є найефективнішою;

1) дослідити ключові принципи мікро сервісної архітектури;

2) визначити, як мікро сервіси можуть забезпечити масштабованість та продуктивність;

3) розглянути виклики, з якими стикаються компанії, що впроваджують мікросервіси [1].

Основи мікросервісної архітектури

Мікросервіси – це підхід до розробки програмного забезпечення, в якому додаток розбивається на набір невеликих незалежних сервісів, кожен з яких виконує окремі функції.

Основні принципи:

- Кожен сервіс має чітко визначену функціональність.
- Сервіси взаємодіють через стандартизовані API.
- Кожен мікросервіс незалежно розгортається, тестується та оновлюється.
- Порівняння з монолітною архітектурою: Монолітний підхід об’єднує всі функціональні частини програми в одне ціле, що робить такі системи складними для масштабування, підтримки та оновлення.

Переваги мікросервісної архітектури

Масштабованість:

- Завдяки розділенню додатка на окремі сервіси можна масштабувати тільки ті компоненти, які потребують ресурсів, що забезпечує ефективне використання ресурсів.
- Підтримка горизонтального масштабування кожного сервісу на окремих серверах або контейнерах.

Гнучкість:

- Дає змогу використовувати різні технології, фреймворки, та мови програмування для окремих сервісів, залежно від вимог.

Незалежність компонентів:

- Розгортання та оновлення окремих сервісів відбувається незалежно, що знижує час простою додатка і підвищує швидкість оновлень.

Продуктивність команд:

- Підхід підвищує продуктивність розробницьких команд, які можуть незалежно працювати над окремими мікросервісами, зменшуючи залежності між командами та час узгоджень [2].

Виклики мікро сервісної архітектури

Складність комунікації:

- Мікросервіси вимагають чіткої комунікації між сервісами через API, що призводить до підвищення складності розробки.
- Потрібне використання протоколів комунікації, як-от HTTP / REST або gRPC, для передачі даних.

Моніторинг і підтримка:

- Більша кількість сервісів означає складнішу систему моніторингу та управління. Необхідне впровадження інструментів, як-от Prometheus, Grafana для моніторингу стану кожного сервісу.
- Забезпечення стабільності та швидке виявлення проблем, особливо в умовах постійного оновлення.

Управління даними:

- Розподілені сервіси мають окремі бази даних, що ускладнює синхронізацію даних між ними. Потрібне впровадження механізмів для забезпечення цілісності та консистентності даних.

- Розробка системи управління транзакціями та роботи з даними між мікросервісами.

Безпека:

- Кожен сервіс повинен бути захищений, оскільки множинні точки доступу до API збільшують вразливість системи.
- Впровадження аетентифікації та авторизації для кожного сервісу, наприклад, через OpenID Connect або OAuth2.

Приклади використання мікросервісної архітектури в клієнт-серверних системах

Netflix:

Завдяки мікросервісній архітектурі Netflix змогла досягти високої масштабованості та стабільності, обробляючи мільярди запитів щодня.

Реалізація незалежних сервісів для кожної функції (рекомендації, пошук, авторизація тощо) дала змогу швидко оновлювати та масштабувати окремі компоненти.

Uber:

Система Uber використовує мікросервіси для забезпечення швидкого обслуговування замовлень та координації водіїв і пасажирів.

Архітектура знижує затримки та допомагає обробляти великий потік запитів у реальному часі.

Amazon:

Впровадження мікросервісів допомогло Amazon забезпечити стабільну роботу інтернет-магазину та надає можливість команді розробників оновлювати окремі сервіси, не впливаючи на загальну роботу системи [3].

Мікросервісна архітектура пропонує ефективний підхід до розробки масштабованих клієнт-серверних систем, що дає змогу забезпечити швидкий розвиток додатків та легкість оновлення.

Основні переваги: висока гнучкість, можливість індивідуального масштабування сервісів, покращення продуктивності команд, незалежне розгортання та оновлення.

Головні виклики: забезпечення надійної комунікації, управління даними, безпека і складність моніторингу.

Рекомендується використовувати мікросервіси для складних, великомасштабних проєктів, де важлива масштабованість і часті оновлення.

Інструменти для управління системою: використання контейнеризації (Docker) та оркестрації (Kubernetes) для спрощення управління мікросервісами.

Забезпечення безпеки та моніторингу: важливо приділяти увагу захисту API та впровадженню систем моніторингу для кожного сервісу.

Список використаних джерел

1. Taibi D., Lenarduzzi V., Pahl C. Architectural Patterns for Microservices: A Systematic Mapping Study. *Journal of Systems and Software.* DOI: 10.1016/j.jss.2017.03.065.

2. Alshuqayran N., Ali N., Evans R. A Systematic Mapping Study in Microservices Architecture. *Journal of Systems and Software.* DOI: 10.1016/j.jss.2017.03.065.

3. Taibi D., Lenarduzzi V., Pahl C. Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation. *IEEE Software. – IEEE Xplore.* DOI: 10.1109/MS.2017.41212.

Наукове видання

Колектив авторів

Матеріали

V ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ «КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ОБРОБКИ ДАНИХ»

Редактор О. А. Солдатова
Технічний редактор Т. О. Важеніна-Гопрак

Підписано до друку 16.12.2024.
Формат 60×84/16. Папір офсетний.
Друк – цифровий. Умовн. друк. арк. 9,76.
Тираж 30. Зам. 99.

Донецький національний університет імені Василя Стуса
21021, м. Вінниця, 600-річчя, 21
Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру
серія ДК № 5945 від 15.01.2018