Міністерство освіти і науки України
Донецький національний університет імені Василя Стуса
Київський національний університет будівництва і архітектури
Львівський національний університет імені Івана Франка
Тернопільський національний технічний університет імені Івана Пулюя

ПРИКЛАДНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Матеріали

VII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ ТА МОЛОДИХ ВЧЕНИХ

15 травня 2026 року

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

Вінниця
2026

Рекомендовано до друку
вченою радою факультету інформаційних і прикладних технологій
Донецького національного університету імені Василя Стуса
(протокол № 14 від 23.06.2026)

Організаційний комітет конференції:

Голова оргкомітету:

СЕГЕДА Сергій Андрійович, доктор економічних наук, професор, в. о. декана факультету інформаційних і прикладних технологій ДонНУ імені Василя Стуса.

Заступники голови комітету:

ВЕСЕЛОВСЬКА Наталія Ростиславівна, доктор технічних наук, професор, в. о. завідувача кафедри інформаційних технологій, заступник декана з наукової роботи ДонНУ імені Василя Стуса.

Відповідальний секретар оргкомітету:

БАБАКОВ Роман Маркович, доктор технічних наук, доцент, професор кафедри інформаційних технологій ДонНУ імені Василя Стуса.

Члени комітету:

НІКОЛЮК Петро Карпович, доктор фізико-математичних наук, професор, професор кафедри інформаційних технологій ДонНУ імені Василя Стуса;

РОТШТЕЙН Олександр Петрович, доктор технічних наук, професор, професор кафедри інформаційних технологій ДонНУ імені Василя Стуса;

ШТОВБА Сергій Дмитрович, доктор технічних наук, професор, професор кафедри інформаційних технологій ДонНУ імені Василя Стуса;

ПОТАПОВА Надія Анатоліївна, кандидат економічних наук, доцент, доцент кафедри інформаційних технологій ДонНУ імені Василя Стуса;

АНТОНОВ Юрій Сергійович, кандидат фізико-математичних наук, доцент, доцент кафедри інформаційних технологій ДонНУ імені Василя Стуса;

БАЦЕНКО Тетяна Василівна, кандидат технічних наук, доцент, доцент кафедри інформаційних технологій ДонНУ імені Василя Стуса;

МОСКВІН Олексій Михайлович, кандидат технічних наук, доцент кафедри інформаційних технологій ДонНУ імені Василя Стуса;

ФРИЗ Ірина Василівна, кандидат фізико-математичних наук, старший викладач кафедри інформаційних технологій ДонНУ імені Василя Стуса;

ПОРЕМСЬКИЙ Юрій Віталійович, кандидат технічних наук, старший викладач кафедри інформаційних технологій ДонНУ імені Василя Стуса;

ПАВЛЮК Олександр Анатолійович, кандидат технічних наук, старший викладач кафедри інформаційних технологій ДонНУ імені Василя Стуса.

Прикладні інформаційні технології: матеріали VII Всеукраїнської науково-практичної конференції здобувачів вищої освіти та молодих вчених (15 травня 2026 р.). Вінниця. ДонНУ імені Василя Стуса, 2026. 196 с.

У збірнику розміщено матеріали VII Всеукраїнської науково-практичної конференції здобувачів вищої освіти та молодих вчених, яка відбулася 15 травня 2026 року. У збірнику висвітлено актуальні питання, що стосуються сучасних викликів вітчизняної та світової науки, інноваційних розробок, а також перспективних напрямів досліджень. Конференція спрямована на активізацію наукового діалогу між молодими дослідниками, обмін ідеями та досвідом, а також сприяння розвитку міждисциплінарних досліджень.

Для науковців, викладачів закладів вищої освіти, а також аспірантів і здобувачів технічних факультетів.

ISSN (print): 2708-5821

© Колектив авторів, 2026
© ДонНУ імені Василя Стуса, 2026

ЗМІСТ

СЕКЦІЯ 1
ПРИКЛАДНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ОРГАНІЗАЦІЙНИХ, СОЦІАЛЬНО-ЕКОНОМІЧНИХ СИСТЕМАХ

Вершигора Р. О., Потапова Н. А.
ВПЛИВ ОБЧИСЛЮВАЛЬНИХ ПОХИБОК У ФІНАНСОВИХ РОЗРАХУНКАХ ТА УПРАВЛІННІ ТОРГІВЛЕЮ 10
Гуцуляк Д. В., Фриз І. В.
ПРОГРАМНИЙ МОДУЛЬ УПРАВЛІННЯ ВІДНОСИНАМИ З КЛІЄНТАМИ ОНЛАЙН-МАГАЗИНУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ 11
Дубовий С. О., Бабаков Р. М.
РОЗРОБКА ІГРОВИХ ДОДАТКІВ У ЖАНРІ «СТРАТЕГІЯ» 13
Засипка Є. В., Веселовська Н. Р.
ВПЛИВ ШТУЧНОГО ІНТЕЛЕКТУ НА ОСОБЛИВОСТІ ВЕБДИЗАЙНУ ... 15
Матіяш М. В., Фриз І. В.
ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ МЕДИЧНИМ ЦЕНТРОМ З AI-ПОМІЧНИКОМ НА БАЗІ GOOGLE GEMINI 17
Міщанчук В. В., Ніколюк П. К.
ВЕБЗАСТОСУНОК ДЛЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ПРОЦЕСІВ ЩОДОБОВОЇ ОРЕНДИ ЖИТЛОВОЇ НЕРУХОМОСТІ 19
Попик М. І., Бабаков Р. М.
РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ ЕЛЕКТРОНІКИ З ФУНКЦІЄЮ РЕКОМЕНДАЦІЇ ТОВАРІВ 22
Сапожнікова В. Є., Баценко Т. В.
МЕТОДИ ФУНКЦІОНАЛЬНОГО МОДЕЛЮВАННЯ ДЛЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ МОНІТОРИНГУ 26
Семен О. Д., Потапова Н. А.
МЕТОДИ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ КОНТРОЛЮ ЯКОСТІ НА МЕБЛЕВОМУ ВИРОБНИЦТВІ 29
Томай А. О., Веселовська Н. Р.
АНАЛІЗ ПАСАЖИРОПОТОКУ НА ОСНОВІ ДАНИХ ПРОЇЗНИХ КАРТОК У СИСТЕМІ АСООП М. ВІННИЦІ 31
Тронь Д. В., Потапова Н. А.
ОГЛЯД МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ВВП НА ОСНОВІ МАКРОЕКОНОМІЧНИХ ІНДИКАТОРІВ 33
Тронь Д. В., Павлюк О. А.
ОГЛЯД АНСАМБЛЕВИХ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ІДЕНТИФІКАЦІЇ КЛЮЧОВИХ МАКРОЕКОНОМІЧНИХ ДРАЙВЕРІВ ІНФЛЯЦІЇ 35
Юрчук Д. М., Павлюк О. А.
КЛАСТЕРИЗАЦІЯ МЕТЕОРОЛОГІЧНИХ СПОСТЕРЕЖЕНЬ ДЛЯ ВИЯВЛЕННЯ ТИПОВИХ ПОГОДНИХ УМОВ 37
Чемес В. С., Павлюк О. А.
БІБЛІОТЕКИ ТА ФРЕЙМВОРКИ REACT ДЛЯ РОБОТИ З КАРТОГРАФІЧНИМИ ДАНИМИ У ТРАНСПОРТНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ 38

СЕКЦІЯ 2
АЛГОРИТМІЗАЦІЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Гончар А. А., Баценко Т. В.
ПРОЄКТУВАННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ МОБІЛЬНОГО ДОДАТКА ДЛЯ ЧИТАННЯ З ФУНКЦІЄЮ ІНТЕРАКТИВНОГО СЛОВНИКА 42
Грибаніна А. О., Потапова Н. А.
ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ ОБСЛУГОВУВАННЯМ КЛІЄНТІВ САЛОНУ КРАСИ 45
Діброва І. С., Бабаков Р. М.
РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЧНОГО СТВОРЕННЯ ОЗВУЧЕНИХ ВІДЕО 47
Калько Д. Р., Антонов Ю. С.
ПОБУДОВА ШАРУ БІЗНЕС-ЛОГІКИ БАГАТОШАРОВИХ ЗАСТОСУНКІВ НА ОСНОВІ GENERIC КЛАСУ 49
Коваль В. С., Потапова Н. А.
АЛГОРИТМИ СОРТУВАННЯ В МЕТОДАХ ОБЧИСЛЕННЯ: ЕФЕКТИВНІСТЬ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ 52
Когут М. Я., Бабаков Р. М.
СИСТЕМА ВІДДАЛЕНОГО АДМІНІСТРУВАННЯ КОМП'ЮТЕРІВ 54
Коморний Є. М., Фриз І. В.
ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ ПІДПРИЄМСТВА 56
Мисько Б. В., Веселовська Н. Р.
МЕТОДИ ПРОЄКТУВАННЯ МОДУЛЬНОЇ АРХІТЕКТУРИ СЕРВЕРНОЇ ЧАСТИНИ CRM-СИСТЕМИ 58
Підруцький Д. А., Веселовська Н. Р.
АРХІТЕКТУРНІ ОСОБЛИВОСТІ ПОБУДОВИ МАСШТАБОВАНИХ МАРКЕТПЛЕЙСІВ 61
Поліщук О. С., Веселовська Н. Р.
РОЗРОБКА АДАПТИВНОЇ ВЕБПЛАТФОРМИ ПОШУКУ ТА ФІЛЬТРАЦІЇ НЕРУХОМОСТІ НА REACT 62
Попова К. О., Фриз І. В.
РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ У ЖАНРІ ГОЛОВОЛОМКИ 64
Приходський Г. С., Потапова Н. А.
ВИКОРИСТАННЯ РУШІЯ GODOT ДЛЯ РОЗРОБКИ 2D-ІГОР 66
Труханська В. О., Веселовська Н. Р.
АНАЛІЗ СУЧАСНИХ JAVASCRIPT-ФРЕЙМВОРКІВ ДЛЯ ПОБУДОВИ КЛІЄНТСЬКОЇ ЧАСТИНИ ВЕБОРІЄНТОВАНОЇ CRM-СИСТЕМИ 69
Тупіцький В. В., Потапова Н. А.
ПОВЕДІНКОВІ ЕКОНОМІЧНІ ПАТЕРНИ В РОЗРОБКАХ КОМП'ЮТЕРНИХ СТРАТЕГІЧНИХ ІГОР 71
Щербань І. О., Бабаков Р. М.
АЛГОРИТМИ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ЛАНДШАФТІВ ТА СИМУЛЯЦІЇ ЕКОСИСТЕМ 73

СЕКЦІЯ 3
ТЕХНОЛОГІЇ ЗБОРУ, ПРЕДСТАВЛЕННЯ, ОБРОБКИ, ЗБЕРІГАННЯ ІНФОРМАЦІЇ В СУЧАСНИХ ІНФОРМАЦІЙНИХ ТА КОМП'ЮТЕРНИХ СИСТЕМАХ

Бондар А. Б., Потапова Н. А.
ВИКОРИСТАННЯ МЕТОДІВ ОБЧИСЛЕНЬ У ВІДТВОРЕННІ ДИНАМІКИ ПЕРСОНАЖІВ ВІДЕОІГОР 77
Бугайов О. С., Потапова Н. А.
МЕТОД ЦЕНТРАЛЬНОЇ РІЗНИЦІ ЧИСЕЛЬНОГО ДИФЕРЕНЦІЮВАННЯ 78
Вовк В. С., Веселовська Н. Р.
РОЛЬ ТИПОГРАФІКИ В ДИЗАЙНІ ВЕБРЕСУРСІВ 80
Головня С. О., Веселовська Н. Р.
ШТУЧНИЙ ІНТЕЛЕКТ У ВЕБДИЗАЙНІ 82
Даниленко І. С., Потапова Н. А.
МЕТОДИ ПРЯМОКУТНИКІВ І ТРАПЕЦІЙ ДЛЯ ЧИСЕЛЬНОГО ОБЧИСЛЕННЯ ІНТЕГРАЛІВ 83
Діброва І. С., Баценко Т. В.
АНАЛІТИКА ДАНИХ У ВЕБДОДАТКА ЗАСОБАМИ POWER BI ТА EXCEL 85
Житник Є. Т., Потапова Н. А.
АНАЛІЗ ВПЛИВУ ПОЧАТКОВОГО НАБЛИЖЕННЯ НА ЗБІЖНІСТЬ ІТЕРАЦІЙНИХ МЕТОДІВ 88
Засипка Є. В., Павлюк О. А.
ВИКОРИСТАННЯ ЧИСЕЛЬНИХ МЕТОДІВ ОПТИМІЗАЦІЇ У НАВЧАННІ НЕЙРОННИХ МЕРЕЖ 91
Козачок А. О., Баценко Т. В.
СУЧАСНІ BI-РІШЕННЯ ДЛЯ ПІДПРИЄМСТВ: ПОРІВНЯЛЬНИЙ АНАЛІЗ МОЖЛИВОСТЕЙ 93
Кохан Д. Ю., Ніколюк П. К.
РЕАЛІЗАЦІЯ ETL-ПАЙПЛАЙНУ ДЛЯ ІНТЕГРАЦІЇ РІЗНОРІДНИХ ДЖЕРЕЛ ДАНИХ У СИСТЕМАХ БІЗНЕС-АНАЛІТИКИ 95
Маланчук О. В., Бабаков Р. М.
АВТОМАТИЗАЦІЯ ПРОЦЕСІВ ПІДБОРУ ПЕРСОНАЛУ ЗА ДОПОМОГОЮ CRM-СИСТЕМИ 98
Маруняк А. О., Антонов Ю. С.
ІНТЕГРАЦІЯ ПАТЕРНІВ REPOSITORY ТА UNIT OF WORK У DATA ACCESS LAYER-СИСТЕМИ КЕРУВАННЯ РОЗКЛАДОМ ЗАКЛАДУ ВИЩОЇ ОСВІТИ 101
Носалюк О. І., Потапова Н. А.
ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ЧИСЕЛЬНИХ МЕТОДІВ ДЛЯ МОДЕЛЮВАННЯ ФІЗИЧНИХ ПРОЦЕСІВ У ВІДЕОІГРАХ 103
Олексієнко Є. О., Павлюк О. А.
ЗАСТОСУВАННЯ МЕТОДУ ТРАПЕЦІЙ У МОБІЛЬНИХ ДОДАТКАХ ДЛЯ ТРЕНІНГУ ФІЗИЧНОЇ АКТИВНОСТІ 105
Черніхевич О. В., Потапова Н. А.
ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ БУТСТРЕПУ ТА ПЕРЕХРЕСНОЇ ПЕРЕВІРКИ ДЛЯ ОЦІНКИ МОДЕЛЕЙ НАБОРУ ДАНИХ 106

Паламарчук Т. Л., Потапова Н. А.
МЕТОД ДИХОТОМІЇ ЯК СПОСІБ ЧИСЕЛЬНОГО РОЗВ'ЯЗАННЯ РІВНЯНЬ.....109
Пацула А. О., Павлюк О. А.
ОБЧИСЛЮВАЛЬНІ МЕТОДИ РОЗРАХУНКУ НОРМАЛЕЙ ВЕРШИН ДЛЯ АЛГОРИТМІВ ЗГЛАДЖУВАННЯ 111
Самоляк Д. В., Потапова Н. А.
ВИКОРИСТАННЯ ГЛИБОКОГО НАВЧАННЯ (DEEP LEARNING) ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ 113
Хуторна Д. В., Потапова Н. А.
МЕТОДИ ЯКОБІ ТА ЗЕЙДЕЛЯ У РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ.....115
Ярошенко Б. М., Потапова Н. А.
ОПТИМІЗАЦІЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ЗА ДОПОМОГОЮ РЕГУЛЯРИЗАЦІЇ МЕТОДАМИ L1 І L2.....117
Федоришин Б. В., Павлюк О. А.
ЕВОЛЮЦІЯ МЕТОДІВ АВТОМАТИЗАЦІЇ ПЕРЕКЛАДУ УКРАЇНСЬКОЇ ЖЕСТОВОЇ МОВИ.....119

СЕКЦІЯ 4
ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ТА ПРИЙНЯТТЯ РІШЕНЬ

Балюра Б. П., Ніколюк П. К.
АЛГОРИТМИ ТА ПРОГРАМНІ ЗАСОБИ ВИЯВЛЕННЯ АНОМАЛІЙ У ДАНИХ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ.....123
Бевзюк А. Ю., Антонов Ю. С.
ПРОЄКТУВАННЯ БАЗИ ДАНИХ СИСТЕМИ ПЕРСОНАЛЬНОГО КОНТРОЛЮ ХАРЧУВАННЯ 125
Богач Т. О., Фриз І. В.
ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ВИБОРУ КРАЇНИ ПЕРЕБУВАННЯ УКРАЇНСЬКИХ МІГРАНТІВ.....128
Грищишен О. Ю., Веселовська Н. Р.
ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ЗАДАЧ КЛАСИФІКАЦІЇ ПАТЕРНІВ 129
Зигарь Д. Д., Баценко Т. В.
АНАЛІЗ ДАНИХ У СИСТЕМАХ ЛОЯЛЬНОСТІ.....131
Клименко Ю. Ю., Веселовська Н. Р.
ВЕКТОРИЗАЦІЯ ТЕКСТОВИХ ДАНИХ (TF-IDF) ЯК АЛЬТЕРНАТИВА КОЛАБОРАТИВНІЙ ФІЛЬТРАЦІЇ У РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ.....133
Клименко Ю. Ю., Потапова Н. А.
ОЦІНКА ТА МІНІМІЗАЦІЯ ОБЧИСЛЮВАЛЬНИХ ПОХИБОК В АЛГОРИТМАХ МАШИННОГО НАВЧАННЯ 135
Кобець Д. А., Бабаков Р. М.
ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ РИНКОВИХ ДАНИХ ДЛЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ІНВЕСТУВАННЯ ТА ОПТИМІЗАЦІЇ ВИТРАТ 137

Ліхоткіна В. І., Потапова Н. А.
МЕТОДИ СТАТИСТИЧНОГО АНАЛІЗУ РЕЗУЛЬТАТІВ
А/В ТЕСТУВАННЯ У ВЕБСИСТЕМАХ 138
Михайляк М. О., Веселовська Н. Р.
ВИКОРИСТАННЯ PYTHON ДЛЯ АНАЛІЗУ
НЕСТРУКТУРОВАНИХ ДАНИХ 140
Пацула А. О., Веселовська Н. Р.
ІНТЕРАКТИВНИЙ 3D-РЕНДЕРИНГ У ВЕБДИЗАЙНІ 142
Поліщук В. С., Ніколюк П. К.
МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ У СИСТЕМАХ
ПІДТРИМКИ ПРИЙНЯТТЯ ІНВЕСТИЦІЙНИХ РІШЕНЬ
ДЛЯ МАЛОГО БІЗНЕСУ 145
Поліщук О. С., Ніколюк П. К.
АНАЛІЗ ТА ГЕНЕРАЦІЯ ТЕКСТУ. ПЕРЕКЛАДАЧІ 147
Ратушний О. С., Потапова Н. А.
ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ
ДЛЯ ПОШУКУ ОПТИМАЛЬНОГО ШЛЯХУ 150
Русавський О. О., Ротштейн О. П.
ЗАСТОСУВАННЯ НЕЧІТКИХ КОГНІТИВНИХ КАРТ
ДЛЯ ПРОГНОЗУВАННЯ ТА ДІАГНОСТИКИ В УЗАГАЛЬНЕНИХ
ДИНАМІЧНИХ СИСТЕМАХ 152
Слободяник Р. Р., Потапова Н. А.
МЕТОДИ МАШИННОГО НАВЧАННЯ У ПРОГНОЗУВАННІ
ЧАСОВИХ РЯДІВ: ТРАНСФОРМЕРИ ПРОТИ
КЛАСИЧНИХ ПІДХОДІВ 155
Тищенко О. М., Потапова Н. А.
ІНТЕЛЕКТУАЛЬНА СИСТЕМА АНАЛІЗУ БАНКІВСЬКИХ ВИПИСОК
З ВИЯВЛЕННЯМ АНОМАЛЬНИХ ВИТРАТ ТА ПРОГНОЗУВАННЯМ
ФІНАНСОВИХ ПОКАЗНИКІВ 157
Уманська А. В., Ніколюк П. К.
МАТЕМАТИЧНІ МОДЕЛІ, АЛГОРИТМИ ТА МЕТОДИ ШІ
В ТРАНСПОРТНІЙ ЛОГІСТИЦІ 159
Щербина Д. С., Веселовська Н. Р.
МЕТОДИ КЛАСИФІКАЦІЇ СТИЛЮ ШАХОВОГО ГРАВЦЯ
ЯК ОСНОВА ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ 161
Юрчук Д. М., Потапова Н. А.
ВІДНОВЛЕННЯ ПРОПУСКІВ У МЕТЕОРОЛОГІЧНИХ
ЧАСОВИХ РЯДАХ МЕТОДАМИ СТАТИСТИЧНОГО НАВЧАННЯ 163
Яценко В. В., Павлюк О. А.
ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ АГРЕГАЦІЇ
У СИСТЕМАХ КОЛЕКТИВНИХ РЕКОМЕНДАЦІЙ 165

СЕКЦІЯ 5
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

Бурківський О. С., Ніколюк П. К.
РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОЇ ІДЕНТИФІКАЦІЇ ОСІБ
ТА ГЛИБОКОГО АНАЛІЗУ ЗОБРАЖЕНЬ ЗА ВІДКРИТИМИ ДАНИМИ 169

Гапоянц Д. В., Фриз І. В.
АРХІТЕКТУРА СЕРВЕРНОЇ ЧАСТИНИ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ЧИТАННЯ З ІНТЕРАКТИВНИМ СЛОВНИКОМ.....172
Гуцол Н. А., Потапова Н. А.
ІНФОРМАЦІЙНА СИСТЕМА ОБРОБКИ ІНЦИДЕНТІВ ТА ЗАЯВОК ЗВЕРНЕНЬ КЛІЄНТІВ.....174
Заболотна У. К., Веселовська Н. Р.
ПСИХОФІЗІОЛОГІЯ КОЛЬОРУ В TRAVEL-ІНТЕРФЕЙСАХ: ЯК НЕЙРОМАРКЕТИНГ «ПРОДАЄ» ВІДПУСТКУ 176
Клименко А. Р., Баценко Т. В.
АНАЛІЗ ВЕБРЕСУРСІВ НА ОСНОВІ ТЕХНОЛОГІЇ ВІДСТЕЖЕННЯ ПОГЛЯДУ КОРИСТУВАЧА.....178
Коваленко Д. С., Бабаков Р. М.
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ БІБЛІОТЕЧНИМ ФОНДОМ НА ОСНОВІ WEB-ТЕХНОЛОГІЙ ТА POSTGRESQL.....182
Костенко Р. О., Ніколюк П. К.
ПОРІВНЯЛЬНИЙ АНАЛІЗ АСИСТИВНИХ МОБІЛЬНИХ ТЕХНОЛОГІЙ ДЛЯ ОСІБ З ВАДАМИ ЗОРУ: СТАН, ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ 183
Лисак В. О., Потапова Н. А.
МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ НАДАННЯ ПРАВОВОЇ ДОПОМОГИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ 186
Молодченко Д. В., Потапова Н. А.
КРОСПЛАТФОРМОВИЙ ЗАСТОСУНОК ДЛЯ ОРГАНІЗАЦІЇ НАВЧАЛЬНОГО ПРОЦЕСУ.....188
Суліма В. К., Потапова Н. А.
ТЕХНОЛОГІЇ ЗБОРУ ДАНИХ ІЗ СОЦІАЛЬНИХ МЕРЕЖ ДЛЯ ФОРМУВАННЯ ІНФОРМАЦІЙНИХ ПОТОКІВ АНАЛІТИЧНИХ СИСТЕМ.....189
Курбангалієв Д. М., Потапова Н. А.
ІНТЕГРАЦІЯ МЕТОДІВ СТАТИСТИЧНОГО НАВЧАННЯ ДЛЯ КОМПЛЕКСНОГО АНАЛІЗУ ДІЯЛЬНОСТІ ІТ-КОМПАНІЙ: ВІД РЕГРЕСІЇ ДО КЛАСТЕРИЗАЦІЇ.....191

СЕКЦІЯ 1
ПРИКЛАДНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ОРГАНІЗАЦІЙНИХ, СОЦІАЛЬНО-ЕКОНОМІЧНИХ СИСТЕМАХ

УДК 004.422.5.

Вершигора Р. О., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ВПЛИВ ОБЧИСЛЮВАЛЬНИХ ПОХИБОК У ФІНАНСОВИХ РОЗРАХУНКАХ ТА УПРАВЛІННІ ТОРГІВЛЕЮ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вплив обчислювальних похибок у фінансових розрахунках є критичною проблемою сучасних інформаційних систем, особливо в умовах автоматизованої торгівлі та обробки великих обсягів даних. У фінансовій сфері навіть незначні похибки можуть накопичуватися та призводити до суттєвих викривлень результатів – неправильного оцінювання активів, помилок у прогнозуванні або фінансових втрат [1]. Саме тому дослідження природи похибок та способів їх мінімізації є одним зі складників оцінки чисельних методів.

Основною причиною виникнення обчислювальних похибок є обмежена точність представлення чисел у комп'ютері. Похибки поділяються на похибки округлення, відсікання та похибки вхідних даних. У фінансових задачах вони особливо небезпечні, оскільки пов'язані з грошовими значеннями, де навіть незначні відхилення можуть мати реальні наслідки. Наприклад, використання типів із плаваючою комою може призводити до неточного представлення десяткових дробів, що накопичується у великих обчисленнях [2].

З погляду чисельних методів, проблема посилюється під час використання ітераційних алгоритмів. Метод дихотомії, метод Ньютона (дотичних) та метод хорд широко застосовуються для знаходження фінансових показників (наприклад, внутрішньої норми прибутковості). Однак їх ефективність залежить від початкових умов та властивостей функції. Невдалий вибір початкового наближення або наявність похибок у розрахунках може призвести до повільної збіжності або навіть до неправильного результату [3].

Аналогічно, у методах чисельного інтегрування, як-от метод трапецій і метод Сімпсона, похибка апроксимації може впливати на точність оцінки фінансових величин, зокрема під час розрахунку дисконтованих грошових потоків. У задачах чисельного диференціювання похибки ще більш критичні, оскільки обчислення базується на різниці близьких значень, що значно підсилює вплив округлення [1; 4].

Проблема обчислювальних похибок у фінансах має практичні наслідки. У високочастотній торгівлі навіть мікроскопічні помилки можуть впливати на автоматичні рішення алгоритмів. У банківських системах або облікових програмах накопичення похибок може призводити до розбіжностей у балансах. Отже, похибки не є лише теоретичною проблемою, а мають прямий економічний вплив [5].

Для покращення точності обчислень застосовуються різні підходи. По-перше, використання більш точних типів даних (наприклад, decimal замість double) дає змогу зменшити похибки округлення. По-друге, важливим є вибір чисельно стійких алгоритмів, які мінімізують накопичення похибок. По-третє, застосовуються адаптивні методи, що автоматично змінюють крок обчислень для підвищення точності. Додатково використовуються перевірки результатів, порівняння різних методів і тестування точності, що допомагає виявляти та контролювати помилки [4].

Отже, обчислювальні похибки є однією з ключових проблем у фінансових розрахунках. Вони можуть суттєво впливати на результати обчислень і призводити до фінансових ризиків. Використання ефективних чисельних методів, правильний вибір алгоритмів і контроль точності є основними шляхами підвищення надійності фінансових систем.

Список використаних джерел

1. Чисельні методи: навч. посіб. / Л. О. Волонтир, О. В. Зелінська, Н. А. Потапова, І. А. Чіков. Вінниця: ВНАУ. 2020. 322 с.

2. Чисельні методи в комп'ютерних науках / В. А. Андруник, В. А. Висоцька, В. В. Пасічник, Л. Б. Чирун, Л. В. Чирун. Львів: Видавництво «Новий світ – 2000», 2017. 470 с.

3. Floating-point numeric types (C#). URL: https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/floating-point-numeric-types

4. Decimal arithmetic for financial applications. URL: https://speleotrove.com/decimal/decifaq.html

5. Chapra S. C. Applied Numerical Methods with MATLAB for Engineers and Scientists. McGraw-Hill, 2018.

УДК 004.4

Гуцуляк Д. В., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Фриз І. В., канд. фіз.-мат. наук, старший викладач кафедри інформаційних технологій

ПРОГРАМНИЙ МОДУЛЬ УПРАВЛІННЯ ВІДНОСИНАМИ З КЛІЄНТАМИ ОНЛАЙН-МАГАЗИНУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах інтенсивного розвитку цифрової економіки та інформаційних технологій актуальність розробки спеціалізованого програмного забезпечення CRM для e-commerce зумовлена критичною необхідністю персоналізації клієнтського досвіду та автоматизації обробки великих масивів транзакційних даних. Висока конкуренція на ринку вимагає від онлайн-магазинів переходу від реактивного обслуговування до проактивного управління лояльністю, що неможливо без інтегрованих систем збору та аналізу поведінкових патернів споживачів. Великі спеціалізовані системи управління відно-

синами з клієнтами використовувати недоцільно у зв'язку з надлишковістю функцій та їх фокусного застосування для великих обсягів продажів та кількості клієнтів.

Відомі методи розв'язання цієї задачі базуються на поєднанні реляційного моделювання баз даних та методів інтелектуального аналізу даних. Для розробки програмного модуля використовуються методи удосконалення програмних модулів управління відносинами з клієнтами [1], а також готові інформаційні CRM-системи [2–3].

Проєктування логічної структури здійснюється на основі нормалізованих моделей SQL, що забезпечують цілісність зв'язків між сутностями клієнтів, замовлень та товарних позицій. Для аналітичної обробки традиційно застосовується метод RFM-сегментації, який класифікує базу за часом останньої покупки, частотою та монетарною цінністю. У складніших архітектурах використовуються алгоритми некерованого машинного навчання, зокрема кластеризація методом K-середніх, що дає змогу групувати споживачів за багатовимірними векторами ознак.

Структура бази даних базується на принципах третьої нормальної форми, що допомагає уникнути надмірності та забезпечити цілісність транзакцій. Основними таблицями бази даних є таблиця клієнтів, що включає такі поля: ідентифікатор, контактні дані, дата реєстрації та характеристики замовлень (зовнішній ключ клієнта, дата, статус, загальна сума) та їх фільтрації за кожним товаром або групою товарів.

Для оптимізації швидкодії аналітичних зрізів впроваджується денормалізована таблиця-сховище, куди автоматично агрегуються дані після кожної завершеної транзакції. Отримані дані формують аналітичні форми для подальшої роботи з інформацією для удосконалення роботи з клієнтами.

Алгоритм сегментації за методом RFM (Recency, Frequency, Monetary) реалізується на рівні запитів до бази даних. Для кожного клієнта розраховуються індивідуальні показники:

R – різниця між поточною датою та датою останнього замовлення;

F – загальна кількість успішних транзакцій;

M – сума всіх витрат.

Для визначення сегменту клієнтів кожен параметр отримує свій бал, внаслідок чого формуються спеціальні таблиці найбільш привабливих клієнтів (за визначеними факторами – вартість замовлення, стабільних клієнтів, клієнтів з одиночною закупкою, потенційних клієнтів, втрачених клієнтів).

Для поглибленого аналізу в систему інтегрується алгоритм кластеризації K-середніх. Процес починається з нормалізації даних, оскільки суми замовлень та їх кількість мають різні порядки величин. Після обробки алгоритм ітеративно визначає групи клієнтів, мінімізуючи цільову функцію J, яка визначається як сума квадратів помилок всередині кластера:

$$J = \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2,$$

де k – кількість кластерів;

x – вектор ознак клієнта, який належить кластеру S_i;

μ_i – центр кластера S_i.

По суті, визначається квадрат Евклідової відстані між клієнтом та центром його сегмента відносно до середнього арифметичного всіх векторів x. Це дає змогу визначити кластери різного класу клієнтів – лояльних; під загрозою відтоку, а також виявити групи клієнтів, як-от «мисливці за знижками» або «покупці преміум-сегмента».

Отримані дані сегментів можуть бути використані в процесі реалізації автоматизованих маркетингових розсилок, а також для формування сценаріїв персональних діалогів з клієнтами. Результати розрахунків зберігаються в окремому полі таблиці клієнтів, що дає змогу фронтенд-частині CRM миттєво відображати відповідні рекомендації для менеджера.

Результатом дослідження є архітектурне рішення CRM-модуля, що ґрунтується на використанні мікросервісної архітектури та ефективній схемі бази даних з використанням індексованих атрибутів для зменшення часу формування відповідей на аналітичні запити. Впровадження розроблених алгоритмів автоматизованої сегментації дає змогу системі в реальному часі ідентифікувати пріоритетні групи клієнтів та адаптувати маркетингові стратегії під конкретні поведінкові сценарії.

Запропонований метод автоматизації процесів управління відносинами з клієнтами забезпечує високу точність профілювання аудиторії та створює умови для масштабування бізнес-процесів онлайн-магазину. Використання комбінованого підходу до аналізу даних мінімізує похибки під час прогнозування відтоку клієнтів та сприяє зростанню конверсії завдяки релевантності персональних пропозицій.

Список використаних джерел

1. Крилик Л., Шаргало Д. Розширення функціональних можливостей програмного модуля інформаційної технології управління відносинами з клієнтами. *Herald of Khmelnytskyi National University. Technical Sciences.* 2025. Vol. 347(1). P. 156–161. DOI: https://doi.org/10.31891/2307-5732-2025-347-21

2. Salesforce CRM. URL: https://www.salesforce.com/crm/ (дата звернення 09.05.2026).

3. HubSpot CRM. URL: https://www.hubspot.com/products/crm (дата звернення 09.05.2026).

УДК 004.45

Дубовий С. О., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Бабаков Р. М., д-р техн. наук, доцент, професор кафедри інформаційних технологій

РОЗРОБКА ІГРОВИХ ДОДАТКІВ У ЖАНРІ «СТРАТЕГІЯ»

Донецький національний університет імені Василя Стуса, м. Вінниця

Ігрова індустрія є однією з найбільш динамічних сфер сучасних інформаційних технологій. Особливе місце серед комп'ютерних ігор займає жанр

стратегій, який поєднує складні алгоритми керування, штучний інтелект, системи взаємодії між об’єктами та механіки управління ресурсами. Розробка подібних застосунків потребує використання сучасних програмних засобів, здатних забезпечити високу продуктивність, гнучкість та можливість масштабування проєкту. Одним із найбільш популярних інструментів для створення ігрових застосунків є Unity, який широко використовується у сфері GameDev для розробки 2D- та 3D-ігор.

Аналіз сучасних досліджень у сфері розробки комп’ютерних ігор свідчить про активний розвиток технологій створення стратегічних застосунків. Значна частина наукових праць присвячена оптимізації ігрових рушіїв, реалізації алгоритмів пошуку шляху, системам штучного інтелекту та процедурній генерації контенту [3; 6]. Особливістю жанру стратегій є необхідність одночасної обробки великої кількості ігрових об’єктів, що створює додаткове навантаження на систему та вимагає застосування сучасних методів оптимізації. Unity є одним із найбільш популярних рушіїв завдяки підтримці компонентної архітектури, мови програмування C#, систем NavMesh, засобів профілювання та мультиплатформеності [1; 2]. Важливою перевагою рушія є також наявність великої спільноти розробників та Asset Store, що дає змогу прискорити створення прототипів ігрових механік.

Метою роботи є дослідження особливостей розробки ігрових застосунків у жанрі «Стратегія» та аналіз можливостей сучасних технологій GameDev для реалізації стратегічних ігор.

У процесі виконання роботи було досліджено особливості створення стратегічних ігор та можливості рушія Unity для реалізації подібних проєктів (рис. 1). Основну логіку гри реалізовано мовою C#, що дало змогу створити гнучку систему взаємодії між ігровими об’єктами. У межах дослідження реалізовано систему юнітів із різними характеристиками та механіку взаємодії між ними. Для навігації ігрових об’єктів використано засоби NavMesh, що забезпечили автоматичний пошук шляху та уникнення перешкод. Також досліджено питання оптимізації продуктивності шляхом повторного використання об’єктів, оптимізації рендерингу та правильної організації сцени. Отримані результати показали, що Unity дає змогу ефективно створювати стратегічні ігри з великою кількістю об’єктів без суттєвих втрат продуктивності.

Внаслідок виконаної роботи було проаналізовано особливості жанру стратегій та досліджено можливості сучасних засобів GameDev для створення стратегічних ігор. Встановлено, що Unity є ефективним інструментом для реалізації складних ігрових механік завдяки своїй гнучкості, компонентній архітектурі та широкому набору вбудованих засобів. Отримані результати підтверджують доцільність використання Unity для розробки сучасних ігрових застосунків у жанрі «Стратегія».

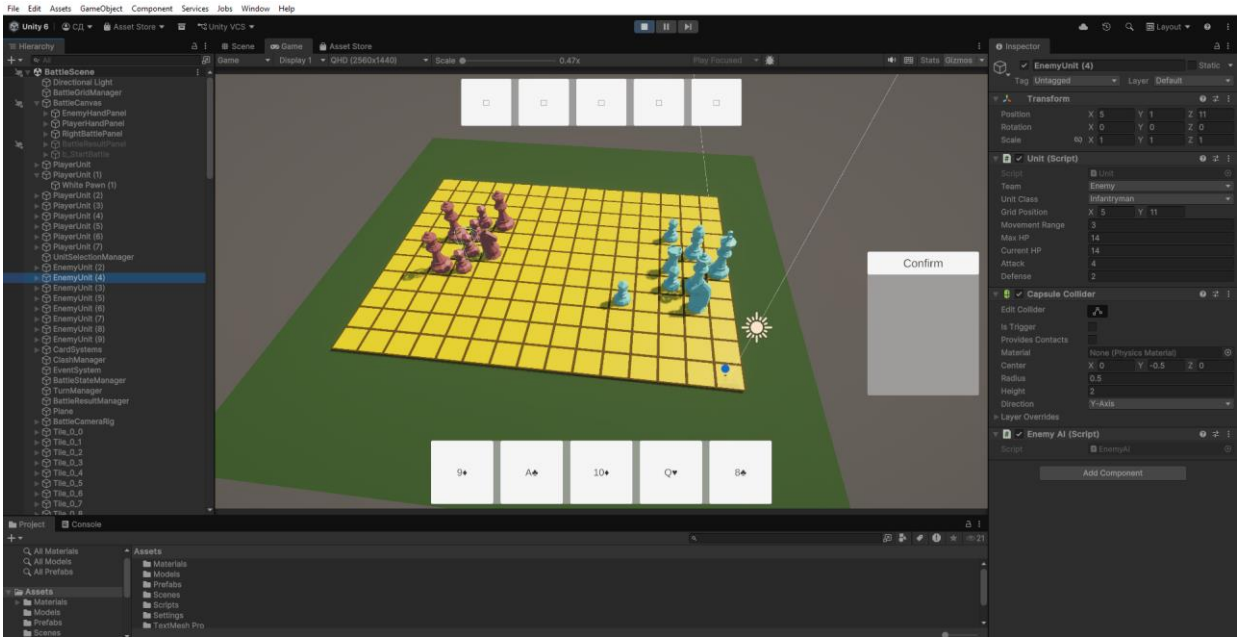


Рисунок 1 – Приклад сцени стратегічної гри, створеної засобами Unity

Список використаних джерел

1. Unity User Manual.
2. Unity Scripting API Documentation.
3. Richard Mansfield. *Mastering Unity Game Development with C#*. Birmingham: Packt Publishing, 2021. 556 p.
4. Will Goldstone. *Unity Game Development Essentials*. Birmingham: Packt Publishing, 2019. 488 p.
5. Millington I., Funge J. *Artificial Intelligence for Games*. CRC Press, 2019. 1136 p.
6. Gregory J. *Game Engine Architecture*. CRC Press, 2018. 1024 p.
7. Nystrom R. *Game Programming Patterns*. Genever Benning, 2014. 354 p.
8. Buckland M. *Programming Game AI by Example*. Jones & Bartlett Learning, 2017. 515 p.

УДК 004.01

Засипка Є. В., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

ВПЛИВ ШТУЧНОГО ІНТЕЛЕКТУ НА ОСОБЛИВОСТІ ВЕБДИЗАЙНУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасний етап розвитку цифрових технологій характеризується фундаментальним переосмисленням ролі штучного інтелекту (ШІ) у процесі проєктування вебінтерфейсів. Якщо раніше ШІ сприймався виключно як інструмент для автоматизації рутинних операцій, то сьогодні він виступає повноцінним суб'єктом дизайну, що змінює саму структуру взаємодії між користувачем та веб-

ресурсом. Головною зміною у вебдизайні під впливом інтелектуальних алгоритмів став перехід від статичного моделювання сторінок до створення живих екосистем, що здатні до самоадаптації.

Ключовим аспектом цього впливу є впровадження концепції гіперперсоналізації. На зміну жорстким шаблонам приходять динамічні інтерфейси, які формуються в реальному часі на основі аналізу великих масивів даних: від історії переходів конкретного користувача до його поточного психоемоційного стану, що зчитується через патерни поведінки. Це фактично скасовує поняття універсального дизайну, замінюючи його мільйонами персоналізованих варіацій одного і того ж ресурсу. Водночас генеративні моделі радикально змінюють швидкість та якість виробництва візуального контенту. Можливість миттєвої генерації унікальної графіки, шрифтових пар та колірних схем дає змогу дизайнерам відмовитися від використання стокових матеріалів, що тривалий час призводило до візуальної уніфікації мережі.

Проте інтеграція ШІ у вебдизайн не обмежується лише естетикою чи зручністю. Вона докорінно змінює архітектуру інформаційного обміну. Поширення розмовних інтерфейсів та нейромережевих пошукових модулів призводить до поступового зникнення класичної навігації. Користувачу більше не потрібно вивчати карту сайта – достатньо сформулювати прямий запит, на який система відповідає не посиланням, а згенерованим рішенням. Це ставить перед дизайнерами нові виклики: на перший план виходить не малювання кнопок, а проєктування логіки взаємодії та мовна модель продукту. Разом із тим, критично важливим стає етичний аспект, адже делегування творчих рішень алгоритмам може призвести до втрати автентичності та виникнення нових форм маніпуляції увагою. Отже, сучасний вебдизайн під впливом ШІ еволюціонує у складну міждисциплінарну галузь, де дизайнер перетворюється з візуального виконавця на архітектора інтелектуальних систем та куратора алгоритмічного досвіду [1–2].

Окремої уваги заслуговує трансформація технічного стеку та методології розробки, де штучний інтелект стає ключовим фактором забезпечення доступності та інклюзивності вебпростору. Завдяки впровадженню інтелектуальних алгоритмів сайти отримують здатність автоматично адаптуватися під потреби людей з обмеженими можливостями: нейромережі в реальному часі генерують змістовні описи для візуального контенту, оптимізують контрастність для користувачів із порушеннями зору та забезпечують безперебійне керування інтерфейсом за допомогою голосу або жестів. Це знімає з дизайнера необхідність вручну налаштовувати сотні параметрів, даючи змогу зосередитися на стратегічному проєктуванні досвіду. Водночас автоматизація тестування та написання коду на основі макетів (Design-to-Code) стирає жорстку межу між графічним редактором і готовим продуктом, що призводить до появи «безшовного» процесу розробки, де ідея втілюється у робочий інтерфейс майже миттєво.

У висновку можна констатувати, що штучний інтелект не замінює вебдизайнера як творчу одиницю, а радикально розширює його когнітивні та технічні можливості, перетворюючи галузь на інтелектуально містку дисципліну. Вплив ШІ проявляється у переході від створення статичних візуальних форм до моделювання складних адаптивних систем, здатних до самонавчання та

глибокої емпатії до потреб користувача. Попри виклики, пов'язані з авторським правом та можливою візуальною одноманітністю алгоритмічних рішень, синергія людського креативу та обчислювальної потужності відкриває шлях до створення нового покоління вебресурсів. Такі ресурси будуть не лише естетично досконалими, а й максимально функціональними, інклюзивними та здатними випереджати запити суспільства, формуючи якісно нове цифрове середовище.

Список використаних джерел

1. Державний стандарт України. (2014). Системи штучного інтелекту. Терміни та визначення. (На заміну ДСТУ 2860–94). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=25034

2. Artificial Intelligence: A Modern Approach – Stuart J. Russell & Peter Norvig, 2020. P. 702–705.

УДК 004.59

Матіяш М. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Фриз І. В., канд. фіз.-мат. наук, старший викладач кафедри інформаційних технологій

ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ МЕДИЧНИМ ЦЕНТРОМ З AI-ПОМІЧНИКОМ НА БАЗІ GOOGLE GEMINI

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасна система охорони здоров'я в Україні переживає активну цифрову трансформацію. Онлайн-системи запису до лікарів стають невід'ємним складником ефективної роботи медичних закладів, забезпечуючи автоматизацію розкладів, зменшення черг та цілодобовий доступ до сервісу. Водночас більшість доступних рішень або не мають засобів первинної допомоги пацієнту у виборі спеціаліста або реалізують їх у вигляді примітивних статичних довідників, що не враховують контексту природномовного опису скарг.

Існуючі популярні платформи, як-от Helsi.me, Health24 та інші, мають суттєві обмеження для невеликих приватних центрів: високі комісії, недостатню гнучкість налаштувань, обмежену інтеграцію з внутрішніми процесами та слабку підтримку додаткових сервісів. Платформи з повноцінною AI-функціональністю (Ada Health, модулі Epic та Cerner) орієнтовані на ринки США та Західної Європи і практично недоступні для вітчизняних закладів [1]. Отже, залишається потреба у власних рішеннях, адаптованих до специфіки роботи українських приватних медичних центрів.

Метою роботи є розробка клієнт-серверного вебзастосунку для медичних центрів, який забезпечує комплексну автоматизацію онлайн-запису на прийом, управління розкладами лікарів та персоналізовані рекомендації через AI-помічник.

З метою вирішення зазначених проблем розроблено вебзастосунок «Ескулап», який побудований на стеку React + Vite + Zustand + Tailwind CSS (фрон-

тенд) та Express + SQLite + JWT + Multer (бекенд). Цей застосунок забезпечує онлайн-запис до лікарів, управління графіками та слотами, персональні кабінети для трьох ролей (пацієнт, менеджер, адміністратор), моніторинг активності, звіти та управління акціями.

Одним із ключових модулів системи є AI-помічник на базі Google Gemini API. Як зазначають А. Д. Топалов та Л. М. Ковальчук [2], інтеграція штучного інтелекту в медичні інформаційні системи дає змогу суттєво підвищити якість взаємодії з пацієнтом та автоматизувати процеси первинної маршрутизації звернень. Архітектура модуля передбачає взаємодію трьох компонентів: клієнтського інтерфейсу чату, серверного проксі-маршруту та зовнішнього Gemini API. Клієнтська частина реалізована у вигляді React-компонента з підтримкою Markdown-розмітки, анімованого індикатора очікування та збереження контексту розмови в межах сесії.

Серверний проксі-маршрут виступає посередником між фронтендом і Gemini API. Він перевіряє JWT-токен користувача, застосовує обмеження частоти запитів (rate limiting – 30 запитів на 15 хвилин), формує системний промпт та передає повідомлення до моделі. Ключ Gemini API зберігається виключно у серверному середовищі, що унеможливлює його розкриття на стороні браузера.

Системний промпт є ключовим механізмом контролю поведінки моделі. Він встановлює для Gemini роль помічника конкретного медичного центру з чіткими обмеженнями: рекомендувати пацієнту відвідати профільного фахівця на основі первинних скарг, а не ставити медичні діагнози або рекомендувати певні препарати. Промпт також містить динамічно згенерований перелік спеціалізацій лікарів, наявних у базі даних, що отримується SQL-запитом піч час кожного звернення. Завдяки цьому модель рекомендує лише реально доступних у закладі спеціалістів і одразу пропонує перейти до онлайн-запису [3].

Для підтримки контексту діалогу весь ланцюг попередніх повідомлень передається до API під час кожного нового запиту. На серверному рівні застосовується обмеження глибини контексту, який становить максимум 20 повідомлень, що запобігає перевищенню ліміту токенів моделі та знижує затримку відповіді.

Функціональне тестування AI-модуля проводилось за 15 сценаріями, що охоплювали типові скарги пацієнтів різних спеціалізацій. В усіх випадках система коректно рекомендувала спеціаліста без постановки діагнозу та згадки про медичні препарати. Наприклад, після отримання скарги «болить горло, температура 38» модуль рекомендував терапевта, а «різкий біль у попереку» – невролога або травматолога. Середній час відповіді становив 1,8–2,4 секунди залежно від довжини контексту діалогу.

Отже, реалізований AI-помічник на базі Google Gemini є ефективним інструментом для прийняття рішенням щодо вибору спеціаліста пацієнтом. До того ж проксі-архітектура із серверною валідацією та системним промптом забезпечує безпеку, контрольованість і медичну доцільність відповідей моделі. Подальшим напрямом розвитку може бути інтеграція з актуальним розкладом лікарів у реальному часі та аналітика частоти звернень за симптомами.

Список використаних джерел

1. Topol E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine.* 2019. Vol. 25. P. 44–56. DOI: 10.1038/s41591-018-0300-7.
2. Топалов А. Д., Ковальчук Л. М. Застосування штучного інтелекту у медичних інформаційних системах. *Вісник НТУУ «КПІ». Серія «Інформатика, управління та обчислювальна техніка».* 2024. № 68. С. 112–119.
3. Google LLC. Gemini API documentation. System instructions and prompting. URL: https://ai.google.dev/gemini-api/docs (дата звернення: 03.05.2026).

УДК 004.4’22:004.738.5

Міщанчук В. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп’ютерні науки,
Ніколюк П. К., д-р фіз.-мат. наук, професор, професор кафедри інформаційних технологій

ВЕБЗАСТОСУНОК ДЛЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ПРОЦЕСІВ ЩОДОБОВОЇ ОРЕНДИ ЖИТЛОВОЇ НЕРУХОМОСТІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Повномасштабне вторгнення росії спричинило безпрецедентні виклики для ринку нерухомості України. Понад 6,5 мільйонів внутрішньо переміщених осіб (ВПО) потребують доступного та безпечного житла [1]. У відносно безпечних регіонах – Вінниці, Хмельницькому, Івано-Франківську, Львові – попит на оренду різко зріс, що призвело до дефіциту доступного житла та зростання цін. Водночас значна частина власників нерухомості бажає надавати житло за зниженими цінами або безкоштовно для тих, хто найбільше постраждав від війни.

Аналіз наявних платформ виявив суттєві недоліки кожної з них. Airbnb – глобальна платформа з зручним інтерфейсом – не має механізму соціальних знижок і не адаптована до українських реалій [2]. OLX Нерухомість не підтримує систему онлайн-бронювання та верифікацію статусу орендарів. Booking.com орієнтована переважно на готелі та має високу вартість підключення [3]. Жодна з платформ не поєднує зручну систему бронювання з верифікованими пільгами для ВПО, військових та волонтерів.

Метою роботи є проєктування та розробка вебплатформи RentFront для короткострокової оренди нерухомості в Україні з вбудованою системою підтримки соціальних ініціатив. Для досягнення мети вирішено такі завдання: аналіз ринку та наявних рішень; формулювання функціональних і нефункціональних вимог; обґрунтування технологічного стеку; проєктування архітектури та бази даних; розробка й тестування платформи з унікальним механізмом верифікованих знижок.

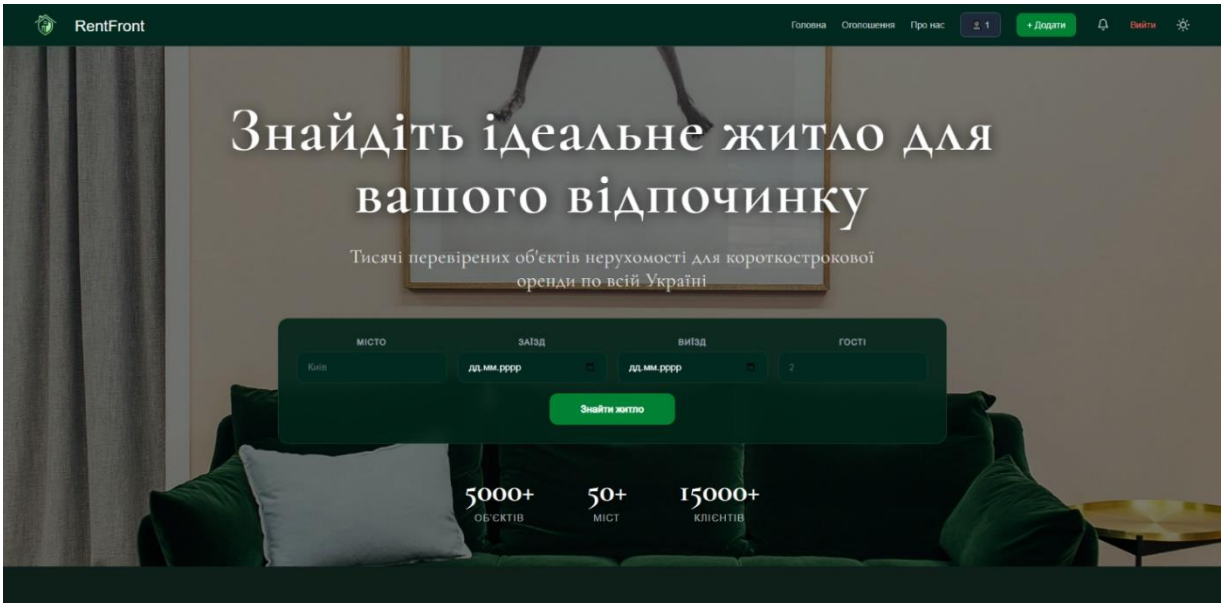


Рисунок 1 – Головна сторінка платформи RentFront

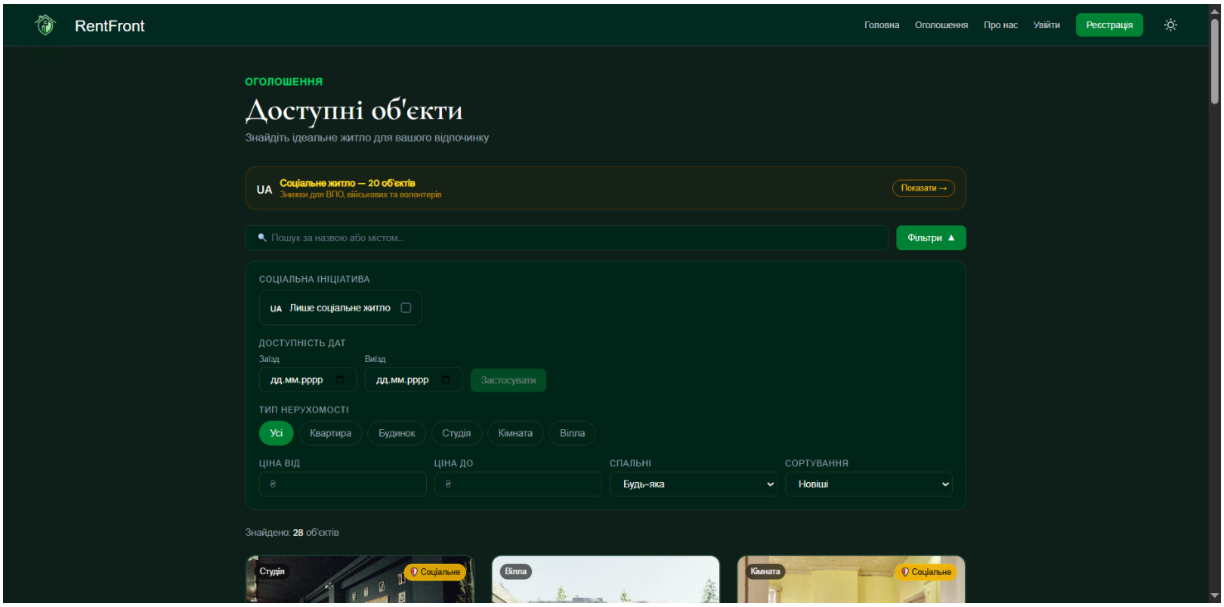


Рисунок 2 – Каталог нерухомості з пошуком та фільтрами

Обґрунтовано вибір технологічного стеку: Next.js 14 з App Router і TypeScript – фронтенд, Cloudflare Workers на основі Edge Computing – бекенд, Cloudflare D1 (SQLite) – база даних, Cloudflare R2 – об’єктне сховище зображень [4; 5]. Така архітектура забезпечує глобальний CDN, мінімальну затримку (Edge Computing), масштабованість без керування серверами та безпеку через JWT-автентифікацію і хешування паролів за алгоритмом PBKDF2.

Платформа реалізує повний набір функцій: каталог нерухомості з пошуком і фільтрацією за типом, ціною, датами, кількістю кімнат та соціальним статусом орендаря; систему бронювання з інтерактивним календарем доступності та автоматичним розрахунком вартості; особистий кабінет та адміністративну панель для модерації оголошень; систему push-сповіщень з лічильником непрочитаних.

Ключовою відмінністю є механізм соціальної ініціативи. Соціальна ініціатива функціонує так: орендодавець активує прапорець «Соціальне житло» та встановлює відсоткові знижки для пільгових категорій – ВПО (discount_displaced), військовослужбовців та ветеранів (discount_military), волонтерів (discount_volunteer). Орендар обирає категорію в особистому кабінеті, після чого адміністратор верифікує статус. Знижка застосовується автоматично: відображається оригінальна (закреслена) та знижена ціна і відсоток пільги.

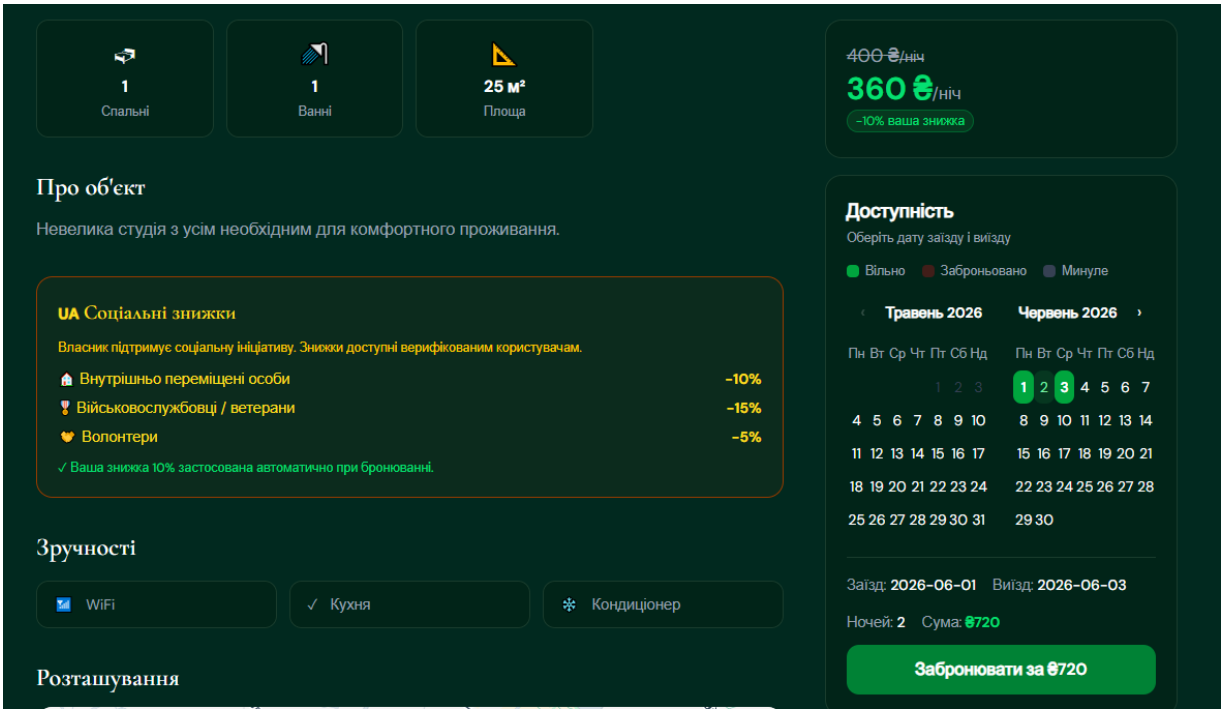


Рисунок 3 – Механізм верифікованих соціальних знижок

Функціональне тестування підтвердило коректну роботу всього основного функціоналу.

Розроблена та розгорнута вебплатформа RentFront (https://rent-front.maffi536.workers.dev) повністю відповідає сформульованим функціональним і нефункціональним вимогам. Платформа є реальним інструментом підтримки ВПО, військових та волонтерів. Перспективи розвитку: інтеграція з Google Maps API, впровадження онлайн-оплати, розробка мобільного додатка на React Native, інтеграція з державними реєстрами для автоматичної верифікації статусу.

Список використаних джерел

1. Ситуація з внутрішньо переміщеними особами в Україні: Моніторингова доповідь МОМ. 2024. URL: https://www.iom.int/ukraine (дата звернення: 01.05.2026).
2. Airbnb. URL: https://www.airbnb.com (дата звернення: 02.05.2026).
3. Booking.com. URL: https://www.booking.com (дата звернення: 02.05.2026).
4. Next.js Documentation. URL: https://nextjs.org/docs (дата звернення: 03.05.2026).
5. Cloudflare Workers Documentation. 2026. URL: https://developers.cloudflare.com/workers/ (дата звернення: 04.05.2026).

УДК 004.42

Попик М. І., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Бабаков Р. М., д-р техн. наук, доцент, професор кафедри інформаційних технологій

РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ ЕЛЕКТРОНІКИ З ФУНКЦІЄЮ РЕКОМЕНДАЦІЇ ТОВАРІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах цифровізації глобальної економіки ринок електронної комерції демонструє стрімке зростання, що зумовлює високу конкуренцію серед онлайн-майданчиків. Користувачі дедалі частіше сприймають інформацію через призму персоналізованого контенту та очікують від вебплатформ інтелектуальної допомоги у виборі технічних пристроїв. Саме тому актуальним напрямом розробки є створення програмних рішень, які дають змогу автоматизувати процес підбору супутніх товарів, формуючи цілісну екосистему навколо обраного бренду та підвищуючи лояльність покупця [1].

Метою роботи є розробка інформаційної системи інтернет-магазину «MyEletroniks», яка поєднує в одному середовищі класичний функціонал електронної торгівлі та кастомний модуль інтелектуальних рекомендацій. Основна особливість запропонованого підходу полягає в тому, що система самостійно аналізує технічну приналежність товару до певної лінійки виробника, автоматично формуючи блоки пропозицій («Більше техніки цього бренду») без необхідності ручного втручання адміністратора для кожної товарної одиниці. На рис. 1 показано головну сторінку інтернет магазину, на рис. 2 показано сторінку каталогу товарів, і на рис. 3–4 показано окремий товар і блоки рекомендацій.

Для реалізації серверної частини вебзастосунку було обрано стек технологій на базі CMS WordPress та плагіна WooCommerce, що забезпечує гнучкість управління базами даних. Програмна логіка розширена за допомогою мови PHP та використання глобального класу WP_Query, який дає змогу організувати складну взаємодію між метаданими товарів та клієнтською частиною [2]. Замість використання стандартних інструментів у системі зареєстровано кастомну таксономію product_brand. Це дало змогу адаптувати видачу під специфіку ніші електроніки: під час отримання запиту система звертається до ідентифікатора бренду, виконує пошук у базі даних та динамічно повертає масив релевантних об'єктів.

Загальна логіка роботи модуля рекомендацій складається з кількох етапів. На першому етапі система ідентифікує унікальні атрибути сторінки, яку переглядає користувач. На другому етапі ініціюється серверний скрипт, який через таксономію брендів визначає ID виробника.

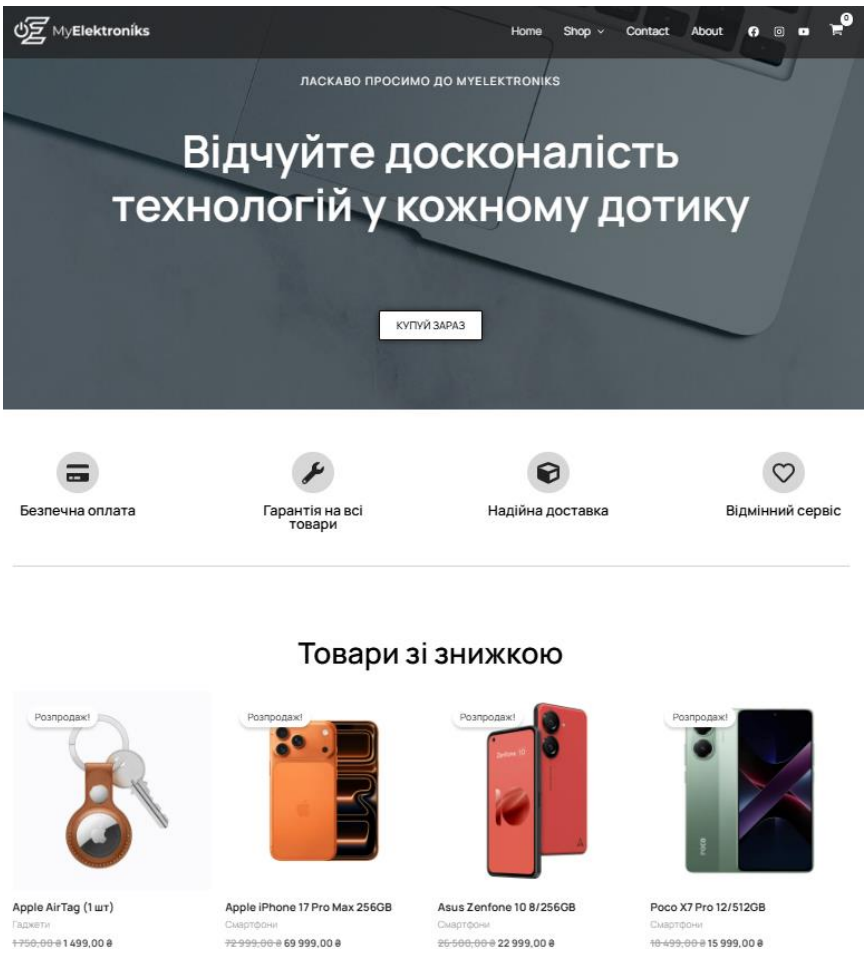


Рисунок 1 – Головна сторінка інтернет-магазину

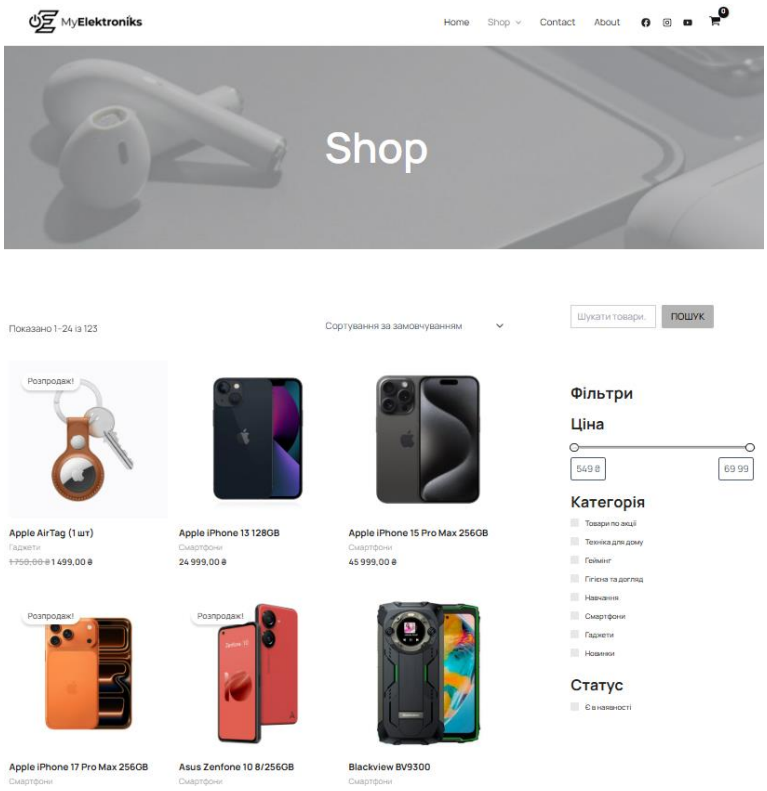


Рисунок 2 – Сторінка каталогу товарів

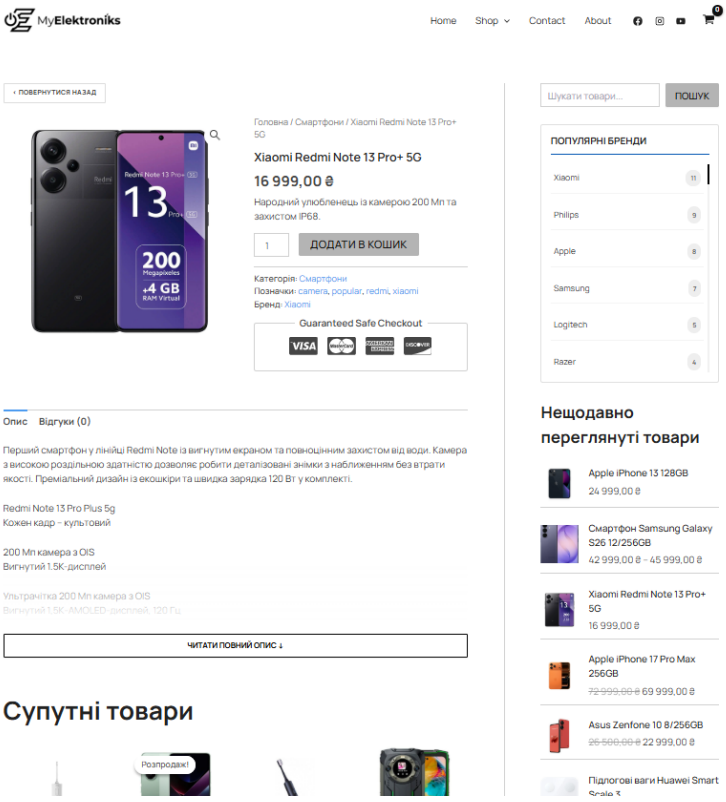


Рисунок 3 – Сторінка товару

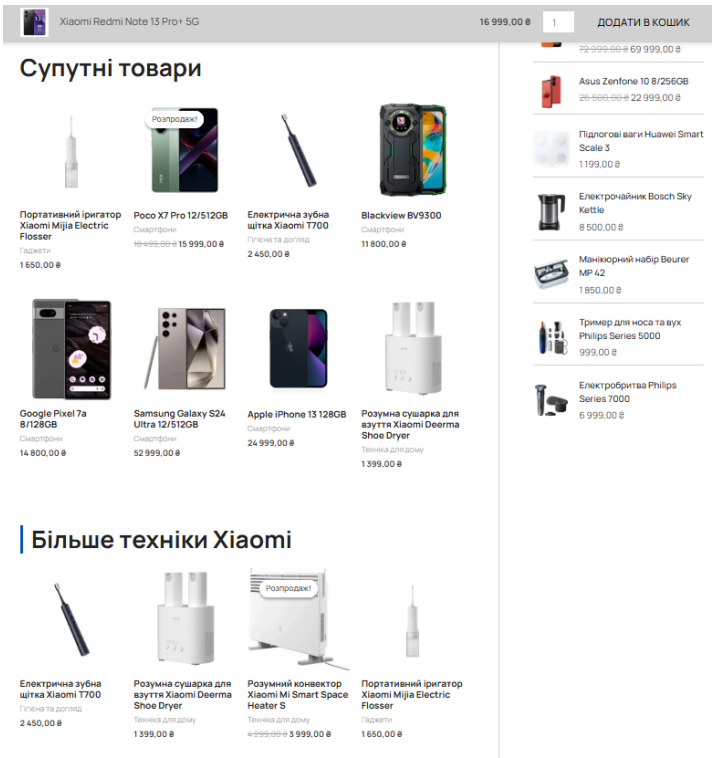


Рисунок 4 – Блоки рекомендацій на сторінці товару

На третьому етапі виконується автоматична генерація блоку рекомендацій із використанням алгоритму випадкового сортування (orderby => rand) та обов'язковим виключенням поточного ID товару (post__not_in), щоб запобігти логічним помилкам дублювання [2]. На завершальному етапі виконується рен-

деринг карток товарів через систему хуків (hooks), що забезпечує візуальну цілісність інтерфейсу.

Перевагою запропонованого підходу є комплексність та висока автоматизація. На відміну від універсальних маркетплейсів, де налаштування зв'язків між товарами часто потребує ручної праці контент-менеджерів, розроблений вебзастосунок орієнтований на автономну роботу в межах заданої логіки бази даних. Це дає змогу власнику бізнесу масштабувати каталог до тисяч найменувань, зберігаючи водночас персоналізований підхід до кожного покупця. До того ж, на відміну від важких зовнішніх скриптів аналітики, це рішення інтегроване безпосередньо в ядро сайта, що мінімізує кількість зовнішніх запитів і позитивно впливає на швидкість завантаження сторінок [3].

Таке рішення має низку практичних переваг. По-перше, користувачеві не потрібно витрачати час на самостійний пошук аксесуарів, сумісних із вибраним гаджетом. По-друге, зменшується кількість рутинних операцій з боку адміністратора сайта. По-третє, вебзастосунок забезпечує повний цикл продажу завдяки інтеграції платіжного шлюзу WayForPay, що дає змогу проводити транзакції через Apple Pay та Google Pay в одному середовищі [4]. По-четверте, використання оптимізованої теми Astra та формату зображень WebP дало змогу досягти показників продуктивності понад 90 балів за версією Google PageSpeed Insights, що є критичним для SEO-просування [3].

Отже, розробка інтернет-магазину з функцією рекомендацій є актуальним напрямом у сфері сучасних вебтехнологій. Запропоноване рішення поєднує технології серверної обробки даних, зручний інтерфейс та інтелектуальні алгоритми вибірки. На відміну від стандартних шаблонів, вебзастосунок «MyEletroniks» забезпечує автоматизоване формування пропозицій, що дає змогу скоротити час підготовки контенту, зменшити технічну складність роботи та зробити процес онлайн-покупок максимально ефективним для споживача [5].

Список використаних джерел

1. Плескач В. Л., Захарченко Г. П. Електронна комерція: підручник. Київ: Знання, 2016. 490 с.

2. WordPress Developer Resources. WP_Query Class Reference. URL: https://developer.wordpress.org/reference/classes/wp_query/ (дата звернення: 12.05.2026).

3. Google Search Central. Core Web Vitals for E-commerce Sites. URL: https://developers.google.com/search/docs/appearance/core-web-vitals (дата звернення: 12.05.2026).

4. Технічна документація інтеграції платіжного модуля. *WayForPay Wiki.* URL: https://wiki.wayforpay.com/ (дата звернення: 12.05.2026).

5. Копитко М. Г., Гірняк В. В. Розробка та проектування інформаційних систем. Львів: ЛьвДУВС, 2021. 248 с.

УДК 004.94:004.415

Сапожнікова В. Є., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Бацєнко Т. В., канд. техн. наук, доцент, доцент кафедри інформаційних технологій

МЕТОДИ ФУНКЦІОНАЛЬНОГО МОДЕЛЮВАННЯ ДЛЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ МОНІТОРИНГУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Проєктування інформаційно-аналітичних систем моніторингу соціальних явищ вимагає чіткої формалізації функціональної архітектури на ранніх етапах розробки. Вибір методу моделювання визначає повноту опису системи, зрозумілість для різних категорій зацікавлених сторін, а також якість подальших проєктних рішень. Від правильно обраного методу залежить те, наскільки точно модель відображає реальні функції системи, її обмеження та взаємодію з зовнішнім середовищем. Порівняльний аналіз наявних підходів до функціонального моделювання дає змогу визначити найбільш відповідний з них для проєктування інформаційно-аналітичних систем моніторингу з урахуванням специфіки їх функціональних вимог.

DFD (Data Flow Diagram) є діаграмною технікою відображення потоків даних, що показує рух інформації між процесами, сховищами та зовнішніми сутностями системи [1]. Метод добре підходить для деталізації інформаційних потоків на рівні реалізації. DFD наочно показує, звідки надходять дані, де вони обробляються і куди передаються.

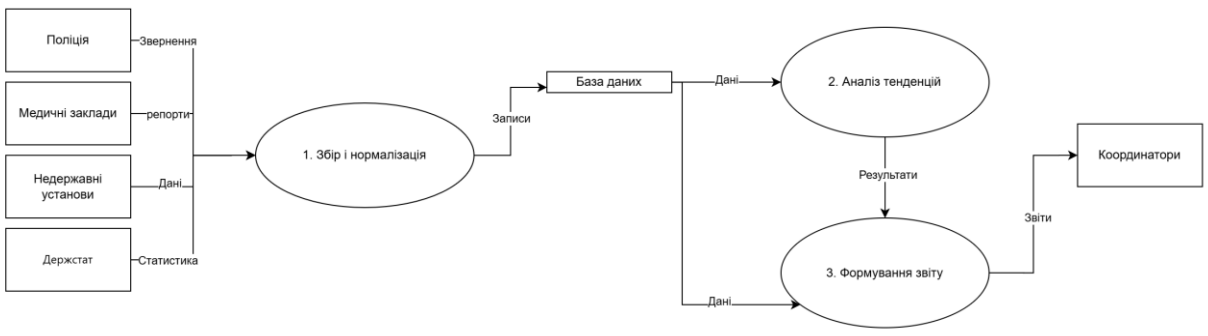


Рисунок 1 – Приклад діаграми потоків даних (DFD) процесу моніторингу соціального явища

На рис. 1 наведено DFD-діаграму системи моніторингу, що включає чотири зовнішні сутності (поліція, медичні заклади, недержавні установи, Держстат), три процеси (збір і нормалізація, аналіз тенденцій, формування звіту) та сховище даних. Попри наочність, DFD не дає змоги явно визначити керувальні впливи та ресурси виконання, що є критичним для систем, які функціонують в умовах нормативного регулювання.

BPMN (Business Process Model and Notation) є стандартом моделювання, орієнтованим на опис виконуваних бізнес-процесів з чітко визначеними учасниками, подіями та потоками управління між ними [2]. Метод розроблявся насамперед для опису організаційної взаємодії між підрозділами та зовнішніми акторами, де ключову роль відіграє послідовність дій виконавців та умови переходу між станами процесу.

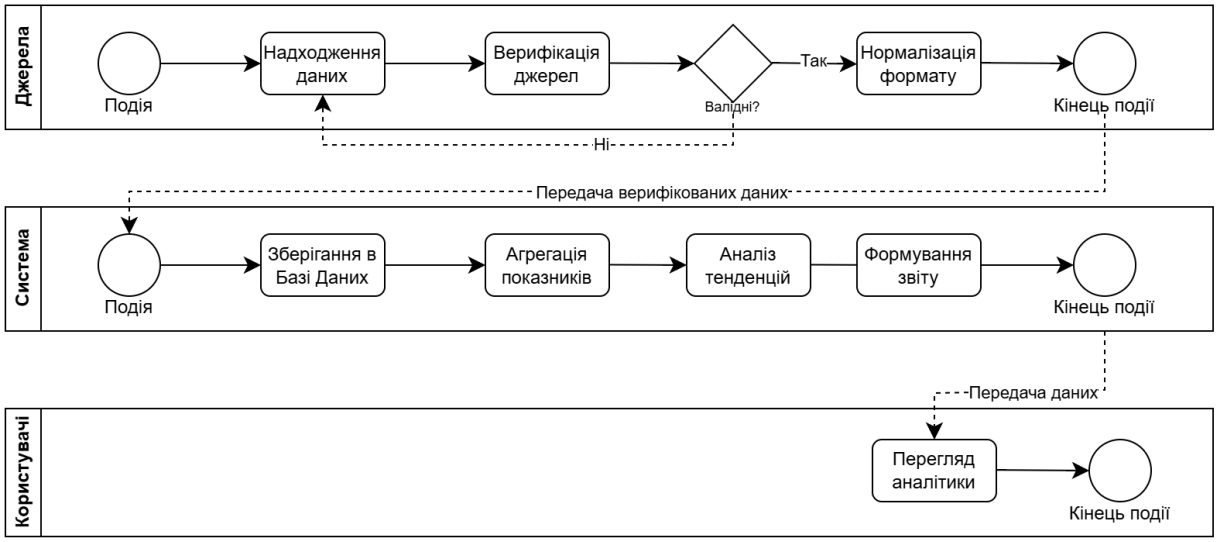


Рисунок 2 – Приклад BPMN-діаграми процесу моніторингу соціального явища

На рис. 2 наведено BPMN-діаграму процесу моніторингу соціального явища, що включає три смуги учасників: джерела даних, інформаційну систему та користувачів. У смузі джерел моделюється надходження та верифікація даних з перевіркою їх валідності через шлюз. Смуга інформаційної системи охоплює зберігання, агрегацію, аналіз та формування звіту. Передача між смугами здійснюється через міжорганізаційні потоки повідомлень. Водночас BPMN не передбачає формального розмежування між вхідними даними, керувальними впливами та механізмами виконання, що обмежує його застосування для концептуального опису функцій аналітичних систем моніторингу.

IDEF0 є методологією функціонального моделювання на основі ієрархічної декомпозиції з використанням нотації ICOM: Input (вхідні дані, що трансформуються), Control (керувальні впливи), Output (результати) та Mechanism (ресурси виконання) [3]. Метод є стандартом IEEE з 1998 р. та призначений для моделювання складних систем на концептуальному рівні, незалежно від деталей технічної реалізації. Моделювання починається з контекстної діаграми A-0 і поступово деталізується через декомпозицію на дочірні функціональні блоки. На рис. 3 наведено контекстну діаграму A-0 та діаграму декомпозиції A0, що містить три функціональні блоки: збір і нормалізація даних, обробка та аналіз, звітування та візуалізація. Входами системи є дані поліції, медичних закладів, NGO та статистичні дані; виходами є аналітичні звіти, карти ризику, групи ризику та рекомендації. Керувальні впливи розподілені між блоками відповідно до їх призначення: законодавство та стандарти збору даних регулюють перший і

другий блоки, державні політики охоплюють третій. Такий розподіл є принципово неможливим у жодному з попередніх методів.

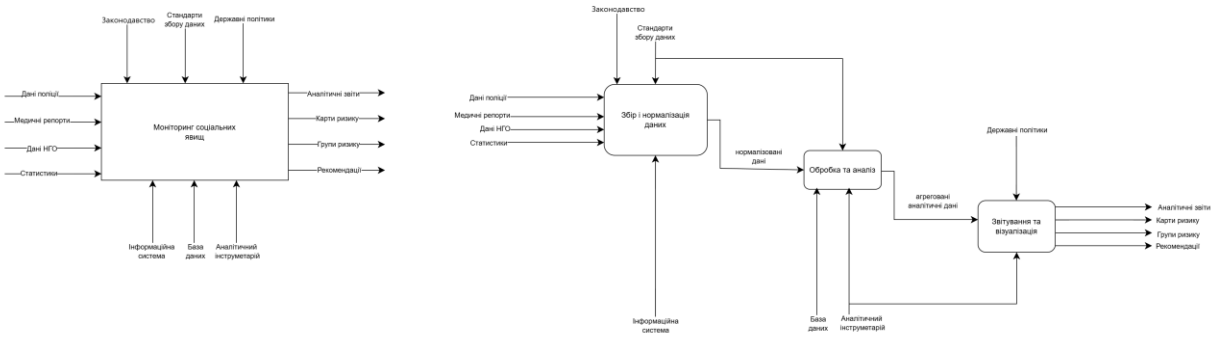


Рисунок 3 – Функціональна модель IDEF0: контекстна діаграма A-0 та діаграма декомпозиції A0

Порівняльний аналіз методів за ключовими критеріями наведено в табл. 1.

Таблиця 1 – Порівняльний аналіз методів функціонального моделювання

Критерій	DFD	BPMN	IDEF0
Моделювання функцій системи	частково	так	так
Явне визначення ICOM	ні	ні	так
Концептуальне проєктування	ні	ні	так
Незалежність від реалізації	так	ні	так
Моделювання потоків даних	так	частково	ні
Моделювання ролей та учасників	ні	так	ні
Ієрархічна декомпозиція	так	ні	так

Результати аналізу свідчать, що IDEF0 є єдиним з розглянутих методів, який одночасно забезпечує чітке визначення функцій через нотацію ICOM, підтримку ієрархічної декомпозиції та незалежність від деталей технічної реалізації. BPMN орієнтований на виконувані процеси з конкретними ролями, що є передчасним на етапі концептуального проєктування. DFD фокусується на потоках даних і не дає змоги формалізувати керувальні впливи. Для інформаційно-аналітичних систем моніторингу, що інтегрують дані з кількох інституційних джерел в умовах нормативного регулювання, метод IDEF0 є оптимальним вибором.

Список використаних джерел

1. Data Flow Diagram. ResearchGate, 2009. URL: https://www.researchgate.net/publication/312836360_Data_Flow_Diagram

2. Chinosi M., Trombetta A. BPMN: An introduction to the standard. Computer Standards & Interfaces. ResearchGate, 2012. Vol. 34. P. 124–134. URL: https://www.researchgate.net/publication/220619834_BPMN_An_introduction_to_the_standard

3. Січко Т. В. Методи моделювання бізнес-процесів підприємства засобами системного аналізу. *Галицький економічний вісник.* 2016. № 2(51). С. 190–201. URL: https://elartu.tntu.edu.ua/handle/ 123456789/20640

4. Міхайленко В. М., Січко Т. В. Моделі і методи автоматизованої системи управління регіонального університетського центру: монографія. Вінниця, ВНАУ, 2014. 184 с.

УДК 004.932:658.562

Семен О. Д., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

МЕТОДИ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ КОНТРОЛЮ ЯКОСТІ НА МЕБЛЕВОМУ ВИРОБНИЦТВІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасна меблева промисловість характеризується переходом до масового виробництва з високим ступенем автоматизації. Використання швидкісних кромкооблицювальних та свердлильно-присадних верстатів із ЧПК дає змогу виготовляти велику кількість деталей за короткий проміжок часу [1]. Проте зростання продуктивності створює серйозний виклик для систем контролю якості. На швидкостях подачі, що характерні для сучасного промислового обладнання, людське око фізично не здатне вчасно помітити дрібні дефекти.

Особливо гостро ця проблема стосується дільниць крайкування меблевих щитів із ДСП та МДФ. Мікроскопічні сколи ламінату, що виникають через знос підрізної пили чи фрези, а також нерівномірне нанесення клейового шва, часто стають помітними лише на етапі збирання готового виробу або вже у кінцевого споживача. Традиційний візуальний контроль оператором є суб'єктивним, залежить від втоми та освітлення в цеху. Саме тому розробка програмних рішень на базі комп'ютерного зору, які могли б у автоматичному режимі аналізувати кожну деталь, є вкрай актуальною задачею для цифровізації меблевої галузі.

Сьогодні найперспективнішим напрямом у сфері автоматичного розпізнавання об'єктів є використання глибоких нейронних мереж [2]. Моделі архітектури YOLO (You Only Look Once) стали галузевим стандартом для завдань, де потрібна висока швидкість обробки зображень. Основна ідея таких мереж полягає у здатності за один прохід обчислити координати об'єкта та його тип. Проте аналіз наявних рішень показує, що більшість універсальних моделей потребують серйозної адаптації для меблевого виробництва. Проблема полягає у складності текстур деревинних плит: природний малюнок дерева може створювати візуальні перешкоди, які алгоритм може помилково прийняти за подряпини або тріщини. До того ж умови промислового цеху (пил, вібрації верстата, відблиски від ламінованих поверхонь) створюють додаткові шуми на зображеннях. Тому основним завданням дослідження є проєктування архітектури системи, яка буде стійкою до таких завад і зможе автоматизувати процес прийняття рішень щодо сортування деталей.

Для побудови системи обрано архітектуру YOLOv8 [3]. Її головною перевагою є відсутність жорстких шаблонів (anchor-free), що дає змогу мережі набагато точніше визначати межі дефектів довільної форми, як-от викривлені подряпини чи викришені шматочки ламінату.

Процес розробки програмного продукту розділений на ключові етапи:

1. Формування бази знань (датасету): оскільки для навчання нейромережі потрібна велика кількість прикладів, планується використання методів аугментації. Це дає змогу на основі обмеженої кількості реальних фотографій створити великий масив даних шляхом цифрового додавання шумів, зміни контрастності та імітації різного ступеня запиленості об'єктива.

2. Розробка пайплайну обробки: зображення з камери спочатку має проходити етап попередньої фільтрації (наприклад, медіанний фільтр для видалення точкових шумів), після чого передаватися до нейромережі для детекції аномалій.

3. Реалізація логічного модуля: це найважливіша частина системи, яка перетворює знайдені нейромережею об'єкти в конкретне управлінське рішення.

Особливу увагу в проєкті приділено алгоритму класифікації виявлених дефектів за ступенем їх критичності. Проста детекція («є дефект» чи «немає») не є достатньою для реального виробництва. Логіка системи має базуватися на аналізі таких параметрів:

– Площа та геометричні розміри: система розраховує фізичний розмір пошкодження. Незначні крапки (менше 0,3 мм²) можуть ігноруватися як допустимі похибки матеріалу.

– Місцезнаходження: якщо дефект виявлено на внутрішній стороні заготовки, яка в майбутньому буде прихована іншою деталлю або задньою стінкою шафи, система класифікує деталь як «придатну».

– Тип аномалії: програма має розрізняти критичні дефекти (сколи ламінату, тріщини) та косметичні (залишки клею, які можна легко видалити).

На основі цих даних алгоритм приймає одне з трьох рішень: пропустити деталь далі, очистити деталь або відправити на повторну обробку, або деталь підлягає утилізації. Така багаторівнева фільтрація дає змогу імітувати роботу досвідченого контролера відділу якості, але з набагато вищою швидкістю та стабільністю.

Автоматизація контролю якості меблевих деталей за допомогою штучного інтелекту є складним, але перспективним завданням. Використання сучасних моделей комп'ютерного зору разом із логічними модулями прийняття рішень дає змогу побудувати систему, яка мінімізує вихід бракованої продукції. Основна перевага такого підходу – це об'єктивність та можливість працювати на швидкостях, недоступних для людини.

Список використаних джерел

1. Lasi H., Fettke P., Kemper H. G. Industry 4.0. *Business & Information Systems Engineering.* 2014. Vol. 6. P. 239–242. (дата звернення: 08.05.2026).

2. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature.* 2015. Vol. 521(7553). P. 436–444. (дата звернення: 08.05.2026).

3. Ultralytics YOLOv8 Docs. 2023. URL: https://docs.ultralytics.com/ (дата звернення: 08.05.2026).

УДК 004.9:656.13

Томай А. О., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук., професор, професор кафедри інформаційних технологій

АНАЛІЗ ПАСАЖИРОПОТОКУ НА ОСНОВІ ДАНИХ ПРОЇЗНИХ КАРТОК У СИСТЕМІ АСООП М. ВІННИЦІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Автоматизована система обліку оплати проїзду (АСООП) є програмно-технічним комплексом, призначеним для обліку транспортних послуг за допомогою електронного квитка, а електронний квиток після реєстрації в такій системі надає пасажиру право на отримання транспортної послуги. У 2025 р. в українському законодавстві також зафіксовано поняття токенізованого електронного квитка як унікального цифрового ідентифікатора на фізичному або електронному носії, що використовується для реєстрації оплати проїзду через АСООП [1; 2].

Актуальність. Актуальність дослідження зумовлена тим, що дані АСООП є одним із найбільш інформативних джерел для оцінювання пасажиропотоку в міському транспорті. На відміну від ручних спостережень, система автоматично фіксує кожну успішну валідацію, що дає змогу отримувати детальну часову та маршрутну картину користування транспортом. Для міста Вінниці це має прикладне значення, оскільки в офіційних бюджетних матеріалах міста прямо враховуються щоденні звіти з АСООП, кількість валідаторів і кількість реєстрацій проїзду в транспорті.

Аналіз останніх досліджень. Аналіз останніх досліджень і нормативних підходів показує, що сучасні системи електронного квитка розглядаються не лише як інструмент оплати, а як джерело даних для транспортної аналітики та управління перевезеннями. Нормативна база України визначає базові терміни АСООП і електронного квитка, а нові технічні вимоги 2025 р. вже описують інтероперабельність засобів реєстрації та токенізований електронний квиток. Це свідчить про перехід від простої системи фіксації оплат до більш універсальної платформи збору й обробки транспортних даних [3].

Мета роботи. Метою роботи є аналіз пасажиропотоку на основі даних проїзних карток, що формуються в АСООП м. Вінниці, та опис технічних особливостей отримання, зберігання і передавання цих даних на бортовому та серверному рівнях.

Постановка завдання полягає в тому, щоб описати принцип роботи системи в умовах реальної експлуатації, визначити склад даних, які фіксуються під час валідації картки, показати роль валідаторів і транспортної консолі в процесі формування журналів подій, а також обґрунтувати, як ці дані можуть бути використані для аналізу пасажиропотоку.

Виклад основного матеріалу. АСООП м. Вінниці функціонує як розподілена система, у складі якої бортове обладнання виконує локальну обробку операцій, а серверна частина приймає та накопичує дані після синхронізації. Практично це означає, що система може працювати без постійного зв'язку, а передавання інформації на сервер здійснюється тоді, коли транспортна консоль отримує доступ до мережі. У Вінниці це реалізовано через мобільний канал передачі даних, а сама консоль додатково містить GPS-модуль, що дає змогу прив'язувати транзакції до руху конкретного транспортного засобу. Для міського транспорту це важливо, оскільки дає технічну основу не лише для обліку оплат, але й для оперативного моніторингу маршрутів.

Ключовим джерелом аналітичної інформації є журнали подій формату EKL, які формуються на валідаторах і на транспортній консолі. У таких журналах фіксуються відкриття та закриття зміни, початок і завершення рейсу, транзакції з картками, службові дії та інші події, що характеризують роботу системи протягом зміни. Для аналізу пасажиропотоку найважливішими є записи про успішні прикладання карток, оскільки кожна така операція відповідає одній поїздці. Під час закриття зміни кожен валідатор формує власний журнал, а консоль формує окремий службовий файл, після чого дані передаються на сервер у повному або частковому обсязі залежно від доступності зв'язку.

У системі використовуються персоніфіковані та неперсоніфіковані картки, а також картки з кількома транспортними продуктами. Валідатор працює з технічними атрибутами носія, а саме з номером картки, UID, серією, станом балансу та параметрами активних продуктів. Така модель дає змогу не зчитувати персональні дані пасажира на рівні пристрою, а лише обробляти службову інформацію, необхідну для тарифікації та обліку. Серія картки визначає тип носія або категорію користувача, а окремі продукти на картці можуть відповідати електронному гаманцю, проїзному або службовим карткам працівників оператора системи.

Логіка списання в АСООП м. Вінниці залежить від типу картки та наявного продукту. Якщо на картці активний діючий проїзний, то поїздка фіксується в межах відповідного продукту без списання коштів з гаманця. Якщо проїзний відсутній або вичерпаний, система переходить до електронного гаманця і списує тарифну суму, яка попередньо завантажена в бортове обладнання. У випадку недостатнього балансу валідатор відображає повідомлення про неможливість оплати. Для банківських карток процес відрізняється тим, що фактичне списання може бути відкладене до моменту серверної обробки транзакції після синхронізації даних.

Із позиції аналізу пасажиропотоку кожне успішне прикладання картки є подією, яку можна інтерпретувати як одну поїздку. На основі масиву таких подій можна визначати інтенсивність перевезень за годинами доби, днями тижня, маршрутами та типами транспорту. Додатково GPS-прив'язка транспорту дає змогу зіставляти часову структуру транзакцій із фактичним положенням транспортного засобу та перевіряти відповідність рейсу заданому маршруту. Саме це робить АСООП не просто системою оплати, а повноцінним джерелом первинних даних для транспортної аналітики [4].

Висновки. АСООП м. Вінниці є інструментом не лише розрахунків за проїзд, а й джерелом структурованих даних для аналізу пасажиропотоку. Робота системи в онлайн та офлайн-режимах, накопичення локальних журналів EKL, використання GPS-даних консолі та обробка карток із різними транспортними продуктами формують інформаційну основу для обліку перевезень. На практиці кожне прикладання картки створює запис, придатний для подальшої агрегації, а отже, – дає змогу будувати оперативну аналітику руху пасажирів у м. Вінниці. Отже, дані проїзних карток у межах діючої АСООП є достатньо повним і технічно надійним джерелом для оцінювання пасажиропотоку.

Список використаних джерел

1. Термін «Автоматизована система обліку оплати проїзду». URL: https://zakon.rada.gov.ua/laws/term/37239%3A78967

2. Термін «Електронний квиток». URL: https://zakon.rada.gov.ua/laws/term/8432%3A1118023; Термін «Токенізований електронний квиток»: https://zakon.rada.gov.ua/laws/term/uk-ro/66963

3. Вінницька міська рада. Паспорт бюджетної програми / показники за АСООП: кількість валідаторів, реєстрацій проїзду, щоденні звіти з АСООП. URL: https://surl.li/behzyj

УДК 004.85:330.43

Тронь Д. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

ОГЛЯД МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ВВП НА ОСНОВІ МАКРОЕКОНОМІЧНИХ ІНДИКАТОРІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Прогнозування валового внутрішнього продукту (ВВП) є критичним інструментом для формування державної економічної політики, бюджетного планування та монетарного регулювання. Традиційні підходи, як-от авторегресійні моделі (ARIMA), векторна авторегресія (VAR) або структурні економетричні моделі, мають добре обґрунтовану теоретичну базу, однак часто не здатні враховувати складні нелінійні взаємозв'язки між великою кількістю макроіндикаторів. У цьому контексті методи машинного навчання пропонують альтернативні або доповнювальні підходи, які дають змогу автоматично виявляти приховані закономірності в даних.

У сучасній літературі методи машинного навчання для макроекономічного прогнозування можна умовно поділити на три категорії: (1) ансамблеві методи на основі дерев рішень (Random Forest, Gradient Boosting, XGBoost, LightGBM); (2) нейромережеві архітектури, зокрема рекурентні нейронні мережі (RNN, LSTM, GRU); (3) методи з регуляризацією (Lasso, Ridge, Elastic Net) та ядерні методи (SVR). Кожна група має свої переваги та обмеження, які розглядаються нижче.

Ансамблеві методи. Random Forest та Gradient Boosting є одними з найбільш популярних завдяки високій інтерпретабельності та стійкості до переобладнання. У роботі [1] проведено порівняльний аналіз ансамблевої моделі градієнтного бустингу (GBR) на реальних даних ВВП України. Дослідження показало, що GBR є потужним інструментом, однак для досягнення максимальної точності потребує ретельного налаштування. Перевагою таких методів є здатність оцінювати важливість кожної вхідної змінної, що дає змогу ідентифікувати ключові макроекономічні драйвери, як-от промислове виробництво, рівень безробіття та експортно-імпортні показники.

Рекурентні нейронні мережі. LSTM та GRU спеціально розроблені для роботи з послідовностями даних, що робить їх природним вибором для макроекономічних часових рядів. Дослідження [2] демонструє успішне використання нейронних мереж для прогнозування макроекономічних показників, зокрема на прикладі України. Ці моделі дають змогу враховувати довгострокові залежності в даних, однак їх недоліками є потреба у великих обсягах історичних даних (щонайменше 20–40 кварталів), чутливість до вибору гіперпараметрів та низька інтерпретабельність отриманих результатів.

Методи з регуляризацією. У випадках, коли кількість макроекономічних індикаторів значно перевищує довжину часового ряду (наприклад, у випадку щоквартальних даних за 10 років та 50+ предикторів), методи Lasso або Elastic Net дають змогу ефективно відбирати значущі змінні та зменшувати ризик переобладнання. Вони є розумним базовим рівнем (benchmark) для порівняння з більш складними моделями.

Для країн з волатильними макропоказниками (наприклад, Україна, Туреччина) особливого значення набуває поєднання класичних економетричних підходів із методами машинного навчання. Як зазначається в роботі [1], це дає змогу підвищити точність прогнозів в умовах нестабільності та структурних змін економіки. Прогнозування на 6–8 кварталів вперед є значно складнішим завданням, оскільки похибки зростають, а різниця між методами часто стає статистично незначущою. Особливу увагу варто приділити короткостроковим методам «nowcasting», що дають змогу оцінювати ВВП у режимі реального часу, та використанню динамічних факторних моделей для адаптації до швидкозмінних умов.

Отже, жоден із методів машинного навчання не є універсальним для прогнозування ВВП. Найкращі практичні результати в умовах української економіки досягаються під час створення ансамблю з кількох алгоритмів (наприклад, XGBoost + LSTM) з адаптивним перемиканням залежно від поточної економічної ситуації. Для державних установ, де важлива пояснюваність моделі, доцільно використовувати методи на основі дерев. Для дослідницьких цілей, де критична найвища точність – рекурентні нейромережі з ретельним налаштуванням гіперпараметрів. Перспективним напрямом є подальше поєднання економічної теорії з гнучкістю методів машинного навчання, а також глибша інтеграція альтернативних джерел даних (вебпошук, супутникові знімки) в прогнозні моделі.

Список використаних джерел

1. Кузьменко О. В. Аналіз та порівняння деяких моделей Data Science в контексті їх застосування для макроекономічного прогнозування. *Вісник Херсонського національного технічного університету.* 2026. № 1(96). С. 260–265. URL: https://journals.kntu.kherson.ua/index.php/visnyk_kntu/article/view/1403
2. Синиціна Ю. П., Кают О. В., Бушуєв К. М. Використання нейронних мереж у прогнозуванні макроекономічних показників підприємства. *Причорноморські економічні студії.* 2019. № 41. С. 126–130. URL: https://bses.in.ua/journals/2019/41_2019/25.pdf
3. Зомчак Л. М., Ракова А. С. Динамічна факторна модель короткострокового прогнозування ВВП України. *Економічний аналіз.* 2018. № 4. С. 17–22. URL: http://nbuv.gov.ua/UJRN/ecan_2018_28%284%29__4

УДК 004.85:336.748

Тронь Д. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук., старший викладач кафедри інформаційних технологій

ОГЛЯД АНСАМБЛЕВИХ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ІДЕНТИФІКАЦІЇ КЛЮЧОВИХ МАКРОЕКОНОМІЧНИХ ДРАЙВЕРІВ ІНФЛЯЦІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

Інфляція є одним із найважливіших макроекономічних показників, що впливає на купівельну спроможність населення, інвестиційний клімат та ефективність монетарної політики. Ідентифікація ключових факторів (рушіїв) інфляції є критичним завданням для центральних банків та урядів. Традиційні економетричні моделі мають добре обґрунтовану теоретичну базу, однак часто не здатні враховувати складні нелінійні взаємозв'язки між великою кількістю потенційних драйверів. У цьому контексті ансамблеві методи машинного навчання пропонують потужний інструментарій для автоматичного відбору найбільш значущих змінних та виявлення прихованих закономірностей.

Метою роботи є систематизація та огляд сучасних ансамблевих алгоритмів машинного навчання (Random Forest, Gradient Boosting, XGBoost) та їх застосування для ідентифікації макроекономічних драйверів інфляції.

В економічній теорії до ключових факторів інфляції традиційно відносять: грошову масу, облікову ставку, рівень безробіття, обмінний курс, індекс цін на імпортовані товари та очікування інфляції. Проблема полягає в тому, що в реальних даних ці фактори сильно корелюють та мають нелінійний вплив. Сучасні дослідження показують, що ансамблеві методи дають змогу автоматично долати ці обмеження. Наприклад, дослідження застосування Random Forest та XGBoost для аналізу інфляції в Албанії виявило, що грошова база та заробітна плата є найвпливовішими факторами. Аналіз рушіїв інфляції в Нігерії з використанням ансамблю з кількох алгоритмів показав, що компонент «їжа»

в індексі споживчих цін є критичним драйвером, а градієнтний бустинг виявився найточнішим алгоритмом.

Варто зазначити, що ансамблеві методи вирішують дві ключові проблеми: моделювання нелінійностей та інтерпретацію результатів. Так, Банк Англії запропонував модель блокового бустингу (Blockwise Boosted Inflation Model), яка поєднує економічну структуру (Phillips curve) з гнучкістю машинного навчання для аналізу нелінійних детермінант інфляції. Дослідження застосування XGBoost в Туреччині підтвердило його ефективність для прогнозування інфляції в умовах економічних криз. В оглядовій роботі «Econometrics of Machine Learning Methods in Economic Forecasting» узагальнюються сучасні підходи до застосування ML в економічному прогнозуванні, включно з ансамблевими методами.

Основні переваги ансамблевих методів у контексті аналізу інфляції – висока точність на основі комбінування кількох алгоритмів, інтерпретабельність через вбудовані механізми оцінки важливості ознак, а також стійкість до викидів та пропусків у даних. Перспективні напрями для України – використання SHAP-аналізу для пояснення нелінійних внесків кожного рушія, включення нестандартних індикаторів (даних інтернет-пошуку, супутникових знімків тощо) та адаптація моделей до роботи в умовах воєнних шоків.

Отже, ансамблеві методи машинного навчання, зокрема Random Forest та XGBoost, є ефективним інструментом для ідентифікації ключових драйверів інфляції. Вони дають змогу автоматично ідентифікувати фактори, враховувати нелінійності та забезпечують високу точність.

Список використаних джерел

1. Buckmann M., Potjagailo G., Schnattinger P. Blockwise Boosted Inflation: Non-linear determinants of inflation using machine learning. *Bank of England.* 2025. № 1. С. 1–42. URL: https://www.bankofengland.co.uk/-/media/boe/files/working-paper/2025/blockwise-boosted-inflation-non-linear-determinants-of-inflation-using-machine-learning.pdf

2. Explainable inflation forecasts by machine learning models. *Expert Systems with Applications.* 2022.

3. Samson T. K., Arya S. Foreign exchange rates, oils price, domestic economic policies, supply chain disruption and inflation rate in Nigeria: Evidence from ensemble learning algorithms. *Journal of Statistics and Operations Research.* 2026. № 15. С. 4523–4539. URL: https://iapress.org/index.php/soic/article/view/2284/1917

4. Basha L., Puka L. Machine learning methods in evaluating the impact of economic factors on the consumer price index in Albania. CITREVJOURNAL. 2024. № 23. С. 5–11. URL: https://crj.cit.edu.al/wp-content/uploads/2014/10/crj-nov-issue-1.pdf

5. Enhancing inflation nowcasting with online search data: a random forest application for Colombia. *Borradores de Economía, Banco de la República (Central Bank of Colombia).* 2025.

УДК 004.032.26

Юрчук Д. М., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук., старший викладач кафедри інформаційних технологій

КЛАСТЕРИЗАЦІЯ МЕТЕОРОЛОГІЧНИХ СПОСТЕРЕЖЕНЬ ДЛЯ ВИЯВЛЕННЯ ТИПОВИХ ПОГОДНИХ УМОВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Метеорологічні дані мають складну багатовимірну структуру. Кожне спостереження одночасно характеризується температурою, тиском, вологістю, швидкістю вітру, кількістю опадів. Аналізувати всі ці параметри окремо незручно, а прямого способу побачити природне групування погодних станів не існує. Кластеризація як один із методів статистичного навчання без учителя дає змогу вирішити це завдання: об'єднати спостереження зі схожими характеристиками в однорідні групи [4, с. 1857–1859].

У практичній реалізації для метеорологічних даних часто використовується метод k-середніх. Його ідея проста: дослідник задає кількість кластерів, а алгоритм намагається мінімізувати відстані всередині груп і максимізувати відстані між ними. Для коректної роботи методу важливо попередньо масштабувати змінні, оскільки тиск, температура та кількість опадів вимірюються в різних одиницях і мають різні діапазони значень [5, с. 78–80]. Без масштабування змінна з більшим числовим діапазоном, наприклад, атмосферний тиск, домінувала б над змінною з меншим діапазоном, як-от кількість опадів, що спотворило б результати кластеризації.

Результатом кластеризації стає розподіл усіх днів спостережень за кількома типами погоди. Наприклад, один кластер може відповідати холодним періодам із низькою температурою та підвищеним тиском, інший – теплим і вологим умовам, третій – перехідним сезонам. Аналіз отриманих груп дає змогу побачити, наскільки рівномірно вибірка покриває різні погодні стани, що важливо для подальшого прогнозного моделювання [6, с. 390–392]. Наприклад, якщо певний тип погоди представлений лише кількома спостереженнями, прогнозна модель для нього буде ненадійною.

Визначення оптимальної кількості кластерів не є тривіальним завданням. Найпоширеніші підходи – метод ліктя та силуетний аналіз – ґрунтуються на внутрішніх метриках якості кластеризації. У багатьох випадках для метеорологічних даних трьох-чотирьох кластерів виявляється достатньо, щоб описати основні сезонні та погодні режими. Застосування кластеризації як допоміжного етапу дає змогу спростити подальший аналіз: замість побудови окремих моделей для кожного дня можна створювати узагальнені моделі для цілих груп подібних погодних умов. Це зменшує обчислювальне навантаження та підвищує стабільність прогнозів, особливо коли вихідний набір даних містить тисячі спостережень.

Список використаних джерел

1. Liao T. W. Clustering of time series data – a survey. *Pattern Recognition.* 2005. Vol. 38, № 11. P. 1857–1874.
2. An Introduction to Statistical Learning: with Applications in R / G. James, D. Witten, T. Hastie, R. Tibshirani. 2nd ed. Springer, 2021. 607 p.
3. Aggarwal C. C. Data Mining: The Textbook. Springer, 2015. 734 p.

УДК 004.42:656.13

Чемес В. С., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук, старший викладач кафедри інформаційних технологій

БІБЛІОТЕКИ ТА ФРЕЙМВОРКИ REACT ДЛЯ РОБОТИ З КАРТОГРАФІЧНИМИ ДАНИМИ У ТРАНСПОРТНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні транспортні інформаційні системи потребують не лише зберігання та обробки даних про маршрути, зупинки й розклад руху, а й зручного візуального представлення цієї інформації для користувача. Одним із найефективніших способів такого представлення є інтерактивна карта, яка дає змогу відображати зупинки, маршрути, напрямки руху та інші просторові об'єкти. У веборієнтованих системах, побудованих із використанням React, для роботи з картами застосовуються спеціалізовані бібліотеки та фреймворки, які забезпечують інтеграцію картографічних сервісів у клієнтський інтерфейс.

React є однією з найпоширеніших бібліотек для створення користувацьких інтерфейсів, оскільки забезпечує компонентний підхід до розробки [1]. У контексті транспортної інформаційної системи це дає змогу створювати окремі компоненти для карти, маркерів зупинок, ліній маршрутів, спливаючих вікон та панелей керування. Завдяки такому підходу інтерфейс системи стає більш гнучким, підтримуваним і придатним до подальшого розширення.

Одним із найпоширеніших рішень для інтеграції карт у React-застосунки є React Leaflet. Ця бібліотека є React-обгорткою над Leaflet і дає змогу працювати з картою через React-компоненти [2]. React Leaflet підтримує додавання маркерів, побудову маршрутів, роботу зі спливаючими вікнами та взаємодію користувача з картою. Важливою перевагою є можливість використання OpenStreetMap як безкоштовної картографічної основи, що робить це рішення особливо доцільним для навчальних та муніципальних транспортних систем.

Іншим популярним рішенням є Mapbox GL JS, який забезпечує створення інтерактивних карт із використанням WebGL [3]. Бібліотека підтримує роботу з великими обсягами геоданих, векторними шарами та розширеною стилізацією карти. Перевагою Mapbox є висока продуктивність і сучасний зовнішній вигляд

карт, однак для невеликих транспортних систем її функціональність може бути надмірною. До того ж повноцінне використання сервісу часто потребує API-ключа та має обмеження безкоштовного тарифу.

Ще одним відомим інструментом для роботи з картами є OpenLayers. Це відкрита JavaScript-бібліотека, яка дає змогу відображати тайли, векторні дані та маркери з різних джерел [4]. Вона підтримує різні типи картографічних шарів і просторових форматів даних, що робить її придатною для GIS-систем. Водночас інтеграція OpenLayers із React є складнішою, порівняно з React Leaflet, а сама бібліотека має вищий поріг входу для розробника.

Окремо варто розглянути Google Maps Platform у поєднанні з React-бібліотеками. Ця бібліотека надає React-компоненти та хуки для роботи з Google Maps JavaScript API [5]. Google Maps має якісну картографічну основу, підтримує маркери, маршрути, геокодування та інші сервіси. Проте для використання Google Maps Platform зазвичай потрібен API-ключ і прив'язка до платіжного акаунта, що може бути недоліком для навчальних або муніципальних проєктів з обмеженим бюджетом.

Для систематизації основних характеристик та визначення доцільності використання картографічних бібліотек у транспортних інформаційних системах було проведено їх порівняльний аналіз, результати якого наведено в табл. 1.

Таблиця 1 – Порівняння бібліотек і фреймворків React для роботи з картами

Критерій	**React Leaflet**	**Mapbox GL JS / react-map-gl**	**OpenLayers**	**Google Maps Platform**
Орієнтація на React	Висока	Висока через React-обгортки	Середня	Висока
Безкоштовність використання	Так, за умови використання OpenStreetMap	Частково, залежить від тарифів Mapbox	Так	Частково, залежить від Google Maps Platform
Складність інтеграції	Низька	Середня	Висока	Середня
Підтримка маркерів зупинок	Так	Так	Так	Так
Підтримка ліній маршрутів	Так	Так	Так	Так
Робота з великими обсягами геоданих	Середня	Висока	Висока	Висока
Гнучкість стилізації карти	Середня	Висока	Висока	Висока
Доцільність для системи розкладу транспорту	Висока	Висока для складних систем	Середня	Середня

Як видно з табл. 1, усі розглянуті рішення можуть бути використані для створення картографічного інтерфейсу транспортної системи. Однак їх доцільність залежить від масштабу проєкту, вимог до функціональності та доступних ресурсів. Для системи відображення розкладу руху громадського транспорту, яка повинна забезпечувати показ зупинок, маршрутів і базову взаємодію корис-

тувача з картою, найбільш раціональним вибором є React Leaflet у поєднанні з OpenStreetMap.

Такий вибір пояснюється кількома чинниками. По-перше, React Leaflet добре інтегрується з компонентною архітектурою React, що відповідає загальній структурі клієнтської частини системи. По-друге, бібліотека дає змогу реалізувати всі необхідні функції: відображення карти, маркерів зупинок, ліній маршрутів і спливаючих інформаційних вікон. По-третє, використання OpenStreetMap як картографічної основи дає змогу уникнути залежності від платних API та зменшити вартість експлуатації системи.

Отже, вибір бібліотеки для роботи з картографічними даними є важливим етапом розробки транспортної інформаційної системи. Використання React Leaflet у поєднанні з OpenStreetMap дає змогу реалізувати зручний картографічний інтерфейс, що забезпечує ефективне відображення маршрутів, зупинок та іншої транспортної інформації у вебзастосунку.

Список використаних джерел

1. React Documentation. React, URL: https://react.dev/
2. React components for Leaflet maps. React Leaflet, URL: https://react-leaflet.js.org/
3. Mapbox GL JS Guide. Mapbox, URL: https://docs.mapbox.com/mapbox-gl-js/guides/
4. OpenLayers Documentation. OpenLayers, URL: https://openlayers.org/doc/
5. React Google Maps Documentation. React Google Maps, URL: https://visgl.github.io/react-google-maps/docs

СЕКЦІЯ 2
АЛГОРИТМІЗАЦІЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.418:004.5

Гончар А. А., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Баценко Т. В., канд. техн. наук, доцент, доцент кафедри інформаційних технологій

ПРОЄКТУВАННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ МОБІЛЬНОГО ДОДАТКА ДЛЯ ЧИТАННЯ З ФУНКЦІЄЮ ІНТЕРАКТИВНОГО СЛОВНИКА

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах мобільні застосунки для читання використовуються не лише для перегляду електронних книг, а й як інструменти для навчання та саморозвитку. Особливої популярності набувають рішення, які поєднують процес читання з можливістю миттєвого отримання перекладу або пояснення слів. Інтерактивний словник дає змогу користувачу швидко працювати з новою лексикою без необхідності переходу до сторонніх сервісів, що робить процес читання більш зручним і безперервним.

Метою роботи є проєктування клієнтської частини мобільного додатка для читання з інтерактивним словником, який забезпечує зручний інтерфейс, персоналізацію процесу читання та підтримку функцій вивчення нових слів.

Для реалізації клієнтської частини застосунку обрано фреймворк Flutter та мову програмування Dart. Використання Flutter надає можливість створити кросплатформний мобільний застосунок із єдиною кодовою базою для Android та iOS, забезпечуючи швидке відображення інтерфейсу та підтримку адаптивного дизайну.

До того ж Flutter передбачає використання єдиної системи віджетів для побудови інтерфейсу, завдяки чому застосунок має узгоджений вигляд на різних платформах, а оновлення функціональності стає простішим [1]. Використання декларативного підходу до побудови інтерфейсу також позитивно впливає на продуктивність та зручність підтримки коду.

У процесі проєктування було визначено основні модулі застосунку: авторизація користувача, бібліотека книг, екран читання, інтерактивний словник, система флешкарток та налаштування інтерфейсу.

На рис. 1 представлено узагальнену архітектуру клієнтської частини мобільного застосунку. Діаграма демонструє взаємодію між UI-екранами, сервісами, моделями даних, локальним сховищем та REST API. Такий підхід забезпечує модульність системи, спрощує підтримку коду та сприяє подальшому розширенню функціональності застосунку [2].

Для збереження даних використовуються окремі моделі, зокрема Book, Bookmark, DictionaryWord, Flashcard та ReadingProgress. Використання окремих моделей даних спрощує підтримку проєкту та забезпечує гнучкість подальшого розвитку застосунку [3].

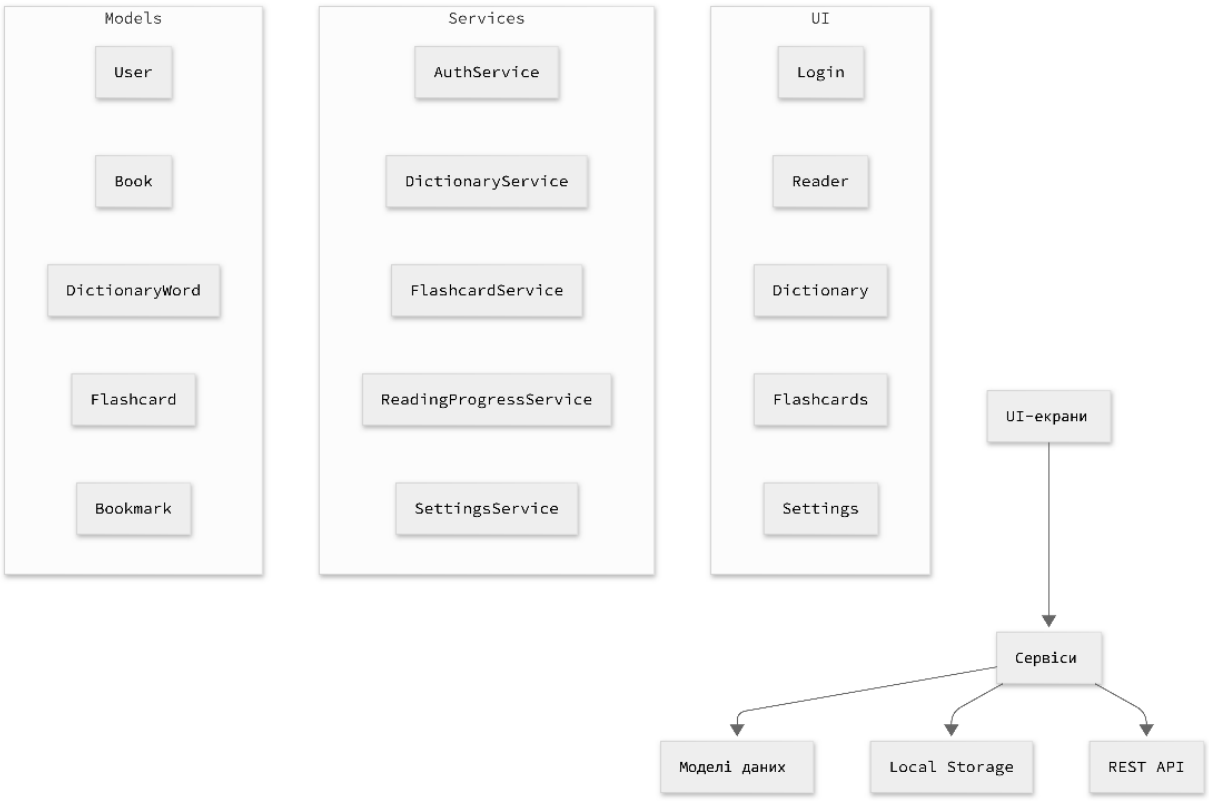


Рисунок 1 – Архітектура клієнтської частини мобільного застосунку

На рис. 2 наведено UML-діаграму класів основних моделей клієнтської частини застосунку. Діаграма відображає взаємозв’язки між основними моделями даних клієнтської частини та демонструє організацію структури даних у застосунку. Використання окремих моделей спрощує організацію даних і забезпечує зручну взаємодію між інтерфейсом користувача та сервісами застосунку [3].

Основним елементом системи є екран читання, у якому реалізовано взаємодію з інтерактивним словником. Користувач може натиснути на слово в тексті, отримати його переклад або пояснення, а також зберегти слово до списку для подальшого повторення. Для покращення користувацького досвіду передбачено підтримку темної теми, зміну розміру шрифту, налаштування кольору фону та збереження прогресу читання [4].

Для оптимізації взаємодії користувача із системою передбачено використання локального збереження налаштувань та кешування окремих даних. Це дає змогу зменшити кількість запитів до сервера, підвищити швидкість роботи застосунку та забезпечити стабільність функціонування навіть за нестабільного інтернет-з’єднання [5].

Інтеграція інтерактивного словника та системи флешкарток створює основу для впровадження елементів мікронавчання у мобільний застосунок. Такий підхід дає змогу поєднати процес читання з поступовим засвоєнням нової лексики та підвищує ефективність самостійного навчання користувачів [2].

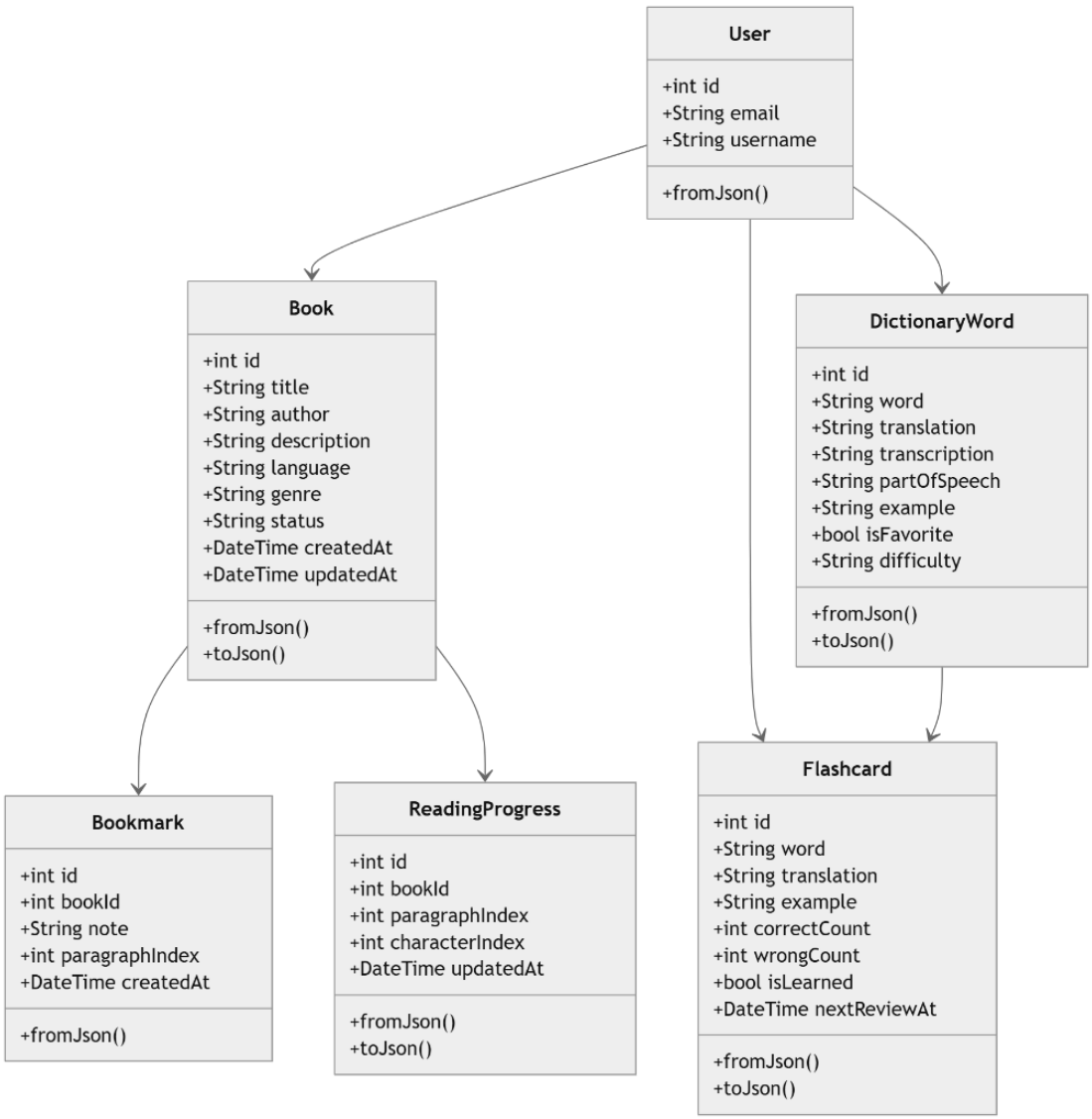


Рисунок 2 – UML-діаграма класів клієнтської частини

Під час аналізу архітектури клієнтської частини було встановлено доцільність використання окремого шару сервісів для взаємодії із серверною частиною та локальним сховищем. Для подальшого розвитку застосунку рекомендовано впровадження сучасних підходів до керування станом, зокрема Provider або Riverpod, що дасть змогу відокремити бізнес-логіку від UI-компонентів та покращити масштабованість системи.

За результатами роботи спроєктовано структуру клієнтської частини мобільного застосунку для читання з інтерактивним словником. Запропоноване рішення забезпечує поєднання функцій читання, персоналізації та підтримки навчання в межах єдиного інтерфейсу. Результати роботи можуть бути використані під час створення мобільних освітніх застосунків та систем електронного читання.

Список використаних джерел

1. Flutter Documentation. URL: https://docs.flutter.dev
2. Understanding user experience for mobile applications: A systematic review / N. Patel et al. *Scientific Reports.* 2025.

3. Android Developers. Guide to app architecture. URL: https://developer.android.com/topic/architecture

4. Material Design Guidelines. URL: https://m3.material.io

5. Експертні та рекомендаційні системи: навч. посіб. / Т. В. Нескородєва, Є. Є. Федоров, Т. В. Січко, А. Р. Нескородєва. Вінниця: ДонНУ імені Василя Стуса, 2022, 208 с.

УДК 004.7

Грибаніна А. О., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ ОБСЛУГОВУВАННЯМ КЛІЄНТІВ САЛОНУ КРАСИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сфера послуг, зокрема індустрія краси, динамічно розвивається завдяки впровадженню цифрових технологій. Сьогодні більшість клієнтів обирає онлайн-запис і шукає оперативний доступ до інформації про доступні послуги. Водночас чимало салонів досі користуються традиційними методами організації запису, що ускладнює процес планування роботи та збільшує навантаження на персонал. Відсутність інформаційних систем спричиняє труднощі з узгодженням розкладу, втрату частини клієнтів і зниження загальної ефективності роботи закладу. Хоча наявні програмні рішення можуть частково розв'язувати ці проблеми, вони часто не відповідають запитам малого та середнього бізнесу через обмежені можливості налаштування або високу вартість. У зв'язку з цим стає актуальним створення вебзастосунку, який враховуватиме особливості функціонування салонів краси та сприятиме підвищенню якості процесів обслуговування клієнтів [1].

Проєкт вебзастосунку має забезпечити ефективну організацію процесу обслуговування клієнтів у салоні краси шляхом автоматизації основних операцій. Основні завдання, що покладено на функціонування вебзастосунку, – це створення ефективного інформаційного середовища для функціонування салону, управління процесом формування замовлень, вибір процедур, комунікація з клієнтами, оформлення робіт з обслуговування клієнтів.

Вебзастосунок створений за клієнт-серверною архітектурою. Інтерфейс користувача реалізовано з використанням сучасних інструментів розробки, що дають можливість створювати адаптивні та інтерактивні вебсторінки. Серверна частина відповідає за обробку запитів, роботу з базою даних і реалізацію основної бізнес-логіки. Особлива увага надана організації взаємодії в режимі реального часу, що забезпечило швидкий та ефективний обмін інформацією між користувачами системи. Для забезпечення безпеки даних впроваджено механізми автентифікації та контролю доступу [5–6].

Створена система реалізує наступні функціональні можливості:

– можливість зручного онлайн-запису для клієнтів;
– організація та ведення графіка роботи майстрів;
– контроль завантаженості та вирішення конфліктів у розкладі;
– обробка платежів;
– обмін повідомленнями між клієнтами й співробітниками;
– налагодження рівнів доступу для різних груп користувачів;
– аналіз діяльності закладу за допомогою візуалізації даних.

Унікальність запропонованого рішення полягає в його комплексному підході до налагодження взаємодії між усіма сторонами процесу. Система гарантує узгодженість даних і автоматичне оновлення інформації, що суттєво мінімізує ймовірність помилок. Головна сторінка користувача вебзастосунку для управління обслуговуванням клієнтів салону краси наведена на рис. 1.



Рисунок 1 – Головна сторінка користувача вебзастосунку для управління обслуговуванням клієнтів салону краси

Впровадження розробленого вебзастосунку сприяє підвищенню ефективності роботи персоналу; зменшенню часу, необхідного для обробки записів; покращенню якості обслуговування клієнтів; раціональнішому використанню робочого часу майстрів; забезпеченню доступу до аналітичних даних для прийняття управлінських рішень.

Отже, проєкт зі створення вебзастосунку спрямований на підвищення загальної ефективності роботи салону через інформатизацію процесів обслуговування клієнтів та зростання якості обслуговування клієнтів.

Список використаних джерел

1. Sommerville I. Software Engineering. Boston: Pearson, 2015. 816 p.
2. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th ed. Boston: Addison-Wesley, 2021. 464 p.
3. Teorey T. J., Lightstone S. S., Nadeau T. Database Modeling and Design. 5th ed. Morgan Kaufmann, 2023. 368 p.

4. WebSocket Protocol: Real-Time Communication for Web Apps. URL: https://www.ietf.org/rfc/rfc6455.txt
5. React Documentation. URL: https://react.dev
6. Socket.IO Documentation. URL: https://socket.io/docs/v4/

УДК 004.8:004.932:004.946

Діброва І. С., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Бабаков Р. М., д-р техн. наук, доцент, професор кафедри інформаційних технологій

РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЧНОГО СТВОРЕННЯ ОЗВУЧЕНИХ ВІДЕО

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах мультимедійний контент активно використовується в освіті, рекламі, соціальних мережах, презентаціях та інформаційній діяльності. Користувачі дедалі частіше сприймають інформацію не лише у текстовому вигляді, а через поєднання зображення, звуку та відеоряду. Саме тому актуальним є створення програмних рішень, які дають змогу швидко перетворювати текстовий матеріал на готовий мультимедійний продукт [1].

Метою роботи є розробка вебзастосунку, який поєднує в одному середовищі автоматичне озвучування тексту та створення відео на основі згенерованої аудіодоріжки й завантажених користувачем зображень. Основна особливість запропонованого підходу полягає в тому, що користувачеві не потрібно окремо використовувати сервіс для синтезу мовлення, завантажувати аудіофайл, відкривати відеоредактор, вручну додавати зображення та синхронізувати їх із голосом [2]. Усі ці дії об'єднуються в єдиному вебзастосунку. На рис. 1 показано головну сторінку вебзастосунку.

Для реалізації функції синтезу мовлення використовується API ElevenLabs, що дає змогу перетворювати текст у природне мовлення, обирати голос і налаштовувати параметри озвучування. До таких параметрів належать стабільність голосу, схожість із вибраним голосовим профілем, стиль мовлення та швидкість відтворення [3]. Це дає змогу адаптувати аудіосупровід під різні типи контенту: навчальні пояснення, інформаційні повідомлення, рекламні ролики або короткі відео для соціальних мереж.

Серверна частина вебзастосунку реалізується за допомогою мови Python та вебфреймворку FastAPI, який використовується для побудови API, обробки запитів користувача та організації взаємодії між клієнтською частиною, сервісом синтезу мовлення і модулем створення відео [4]. Після введення тексту система передає його до сервісу синтезу мовлення, отримує аудіофайл, зберігає його на сервері та використовує для подальшого формування відеофрагмента. Для створення відео застосовується FFmpeg – набір інструментів для обробки аудіо- та відеофайлів, який дає змогу поєднувати зображення з аудіодоріжкою, налаштовувати формат, роздільну здатність і параметри кодування [5].

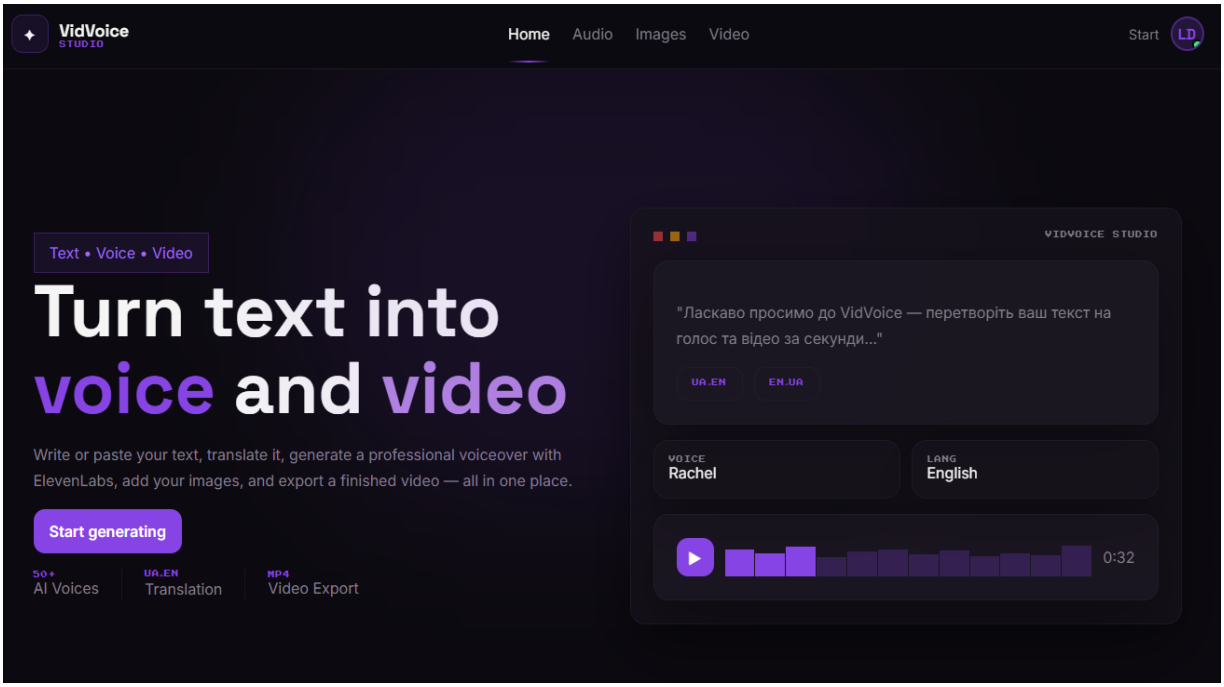


Рисунок 1 – Головна сторінка вебзастосунку для створення озвучених відео

Загальна логіка роботи вебзастосунку складається з кількох етапів. На першому етапі користувач вводить або завантажує текстовий сценарій. На другому етапі виконується генерація аудіо на основі вибраного голосу. На третьому етапі користувач додає зображення, які мають бути використані у відеоряді. Після цього серверна частина автоматично поєднує аудіофайл із зображеннями та формує готовий відеофайл. За потреби окремі відеофрагменти можуть об'єднуватися в один фінальний ролик.

Перевагою запропонованого підходу є комплексність. На відміну від окремих сервісів для генерації мовлення або відеоредакторів, розроблений вебзастосунок орієнтований саме на автоматизацію конкретного сценарію: текст перетворюється на озвучене відео в одному середовищі. Наприклад, ElevenLabs є потужним сервісом для синтезу мовлення, однак його основне призначення полягає саме у створенні аудіо [3]. Canva, VEED або CapCut надають ширші можливості для редагування відео, але вони більше орієнтовані на ручне створення та редагування контенту. У запропонованому рішенні акцент зроблено не на складному монтажі, а на швидкому автоматизованому створенні відео на основі тексту й зображень.

Таке рішення має низку практичних переваг. По-перше, користувачеві не потрібно переходити між різними сервісами та вручну переносити файли. По-друге, зменшується кількість рутинних операцій, пов'язаних із записом голосу, підбором тривалості кадрів, синхронізацією аудіо та зображень. По-третє, вебзастосунок може бути доступним із будь-якого пристрою через браузер, що робить його зручним для використання в навчанні, малому бізнесі, SMM та особистих проєктах. По-четверте, система може бути розширена завдяки додаванню перекладу тексту, збереженню проєктів, вибору мови, керуванню використаними символами для озвучування та повторного редагування створених матеріалів.

Окремо варто зазначити, що автоматичне створення відео на основі текстових або вебматеріалів є актуальним напрямом досліджень. Зокрема, у роботах, присвячених автоматичному формуванню відео, розглядаються підходи до перетворення інформаційного контенту у відеоформат із використанням часових, візуальних і структурних обмежень [6]. Це підтверджує доцільність розробки інструментів, які спрощують створення мультимедійних матеріалів і роблять цей процес доступнішим для користувачів без спеціальної підготовки.

Отже, розробка вебзастосунку для автоматичного створення озвучених відео є актуальним напрямом у сфері сучасних вебтехнологій і мультимедійної обробки даних. Запропоноване рішення поєднує технології синтезу мовлення, серверної обробки аудіо- та відеофайлів і зручний вебінтерфейс. На відміну від окремих сервісів для озвучування або відеомонтажу, вебзастосунок забезпечує повний цикл створення контенту в одному середовищі. Це дає змогу скоротити час підготовки відеоматеріалів, зменшити технічну складність роботи та зробити процес створення озвучених відео доступним для широкого кола користувачів.

Список використаних джерел

1. Planning the development of text-to-speech synthesis systems based on deep learning: a review / H. A. Ahmad, M. Alharbi, A. Alshutayri, S. S. Aljameel. *Journal of King Saud University – Computer and Information Sciences.* 2024. DOI: 10.1016/j.jksuci.2024.102131.

2. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech / Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, T.-Y. Liu. *arXiv preprint.* 2020. DOI: 10.48550/arXiv.2006.04558.

3. ElevenLabs. Text to Speech API Documentation. ElevenLabs Documentation. URL: https://elevenlabs.io/docs/overview/capabilities/text-to-speech

4. FastAPI. FastAPI framework documentation. URL: https://fastapi.tiangolo.com

5. FFmpeg. FFmpeg Documentation. URL: https://ffmpeg.org/documentation.html

6. Automatic Video Creation From a Web Page / P.-Y. Chi, Z. Sun, K. Panovich, I. Essa. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 2020. DOI: 10.1145/3379337.3415814.

УДК 004.415.2

Калько Д. Р., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Антонов Ю. С., канд. фіз.-мат. наук, доцент, доцент кафедри інформаційних технологій

ПОБУДОВА ШАРУ БІЗНЕС-ЛОГІКИ БАГАТОШАРОВИХ ЗАСТОСУНКІВ НА ОСНОВІ GENERIC КЛАСУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Корпоративні застосунки потребують реалізації типових CRUD-операцій (створення, читання, оновлення, видалення) для багатьох сутностей. Наївний підхід із написанням окремого сервісу для кожної сутності призводить до масштабного дублювання коду та суттєво ускладнює підтримку системи.

Актуальність. Зі зростанням кількості доменних сутностей дублювання логіки збільшує технічний борг і ризик помилок. Тому пошук архітектурного рішення для шару бізнес-логіки (BLL), яке уніфікує базові операції, але зберігає гнучкість для специфічних розширень, є надзвичайно актуальним завданням.

Аналіз останніх досліджень. Питання архітектурної організації сервісного шару розглядається у низці робіт. Архітектурні патерни як засіб структурування коду описано у праці Richards [1], де показано переваги стандартизації підходів до розробки багаторівневих систем, що дає змогу делегувати конкретні кроки за умови збереження загального скелету алгоритму. Bogard обґрунтовує концепцію архітектури вертикальних зрізів (Vertical Slice Architecture) як основу гнучкої побудови застосунків, що доповнює теоретичне підґрунтя для підходу, розглянутого у роботі [2]. Питання практичної інтеграції подібних архітектурних рішень на прикладі системи управління розкладом розглянуто в роботах [3; 4]. Синхронну версію CRUD інтерфейсу та GenericCrudRepository розглянуто у роботі [5].

Мета. Метою роботи є дослідження та обґрунтування архітектурного підходу на основі узагальненого сервісного класу для побудови єдиного базового CRUD-сервісу у шарі бізнес-логіки, що забезпечує усування дублювання коду.

Постановка завдання. Необхідно спроєктувати базовий сервісний клас, який: реалізує незмінний алгоритм CRUD-операцій, придатний для довільних доменних сутностей; інтегрує гнучкий механізм валідації бізнес-правил; гарантує атомарність операцій через патерн Unit of Work.

Виклад основного матеріалу. Запропонований підхід базується на трирівневій архітектурі: шар доступу до даних (DAL) реалізує патерни Repository та Unit of Work; шар бізнес-логіки (BLL) містить сервіси, що взаємодіють з DAL виключно через інтерфейси; UI/WebAPI делегує логіку сервісному шару. Структуру взаємодії компонентів наведено на рис. 1.

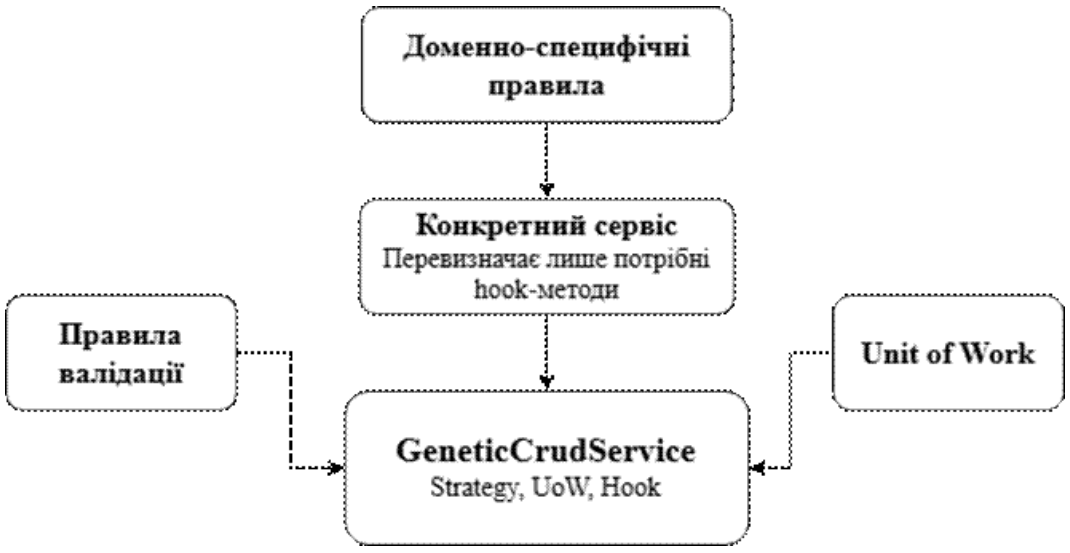


Рисунок 1 – Схема загального сервісного класу

Центральним елементом BLL є узагальнений базовий сервісний клас з двома типовими параметрами – типом DTO та типом доменної сутності. Він реалізує

стандартний інтерфейс сервісу і слугує базовим класом для всіх конкретних сервісів системи, що дає змогу уникнути дублювання коду під час збереження індивідуальної логіки кожного сервісу.

Механізм розширення Generic CRUD Service підкласами реалізовано через віртуальні методи перетворення між сутністю та об'єктом передачі даних, а також через властивість, що повертає типізований репозиторій конкретної сутності. Для простих довідникових сутностей підклас перевизначає лише властивість репозиторію – решту бере на себе базова реалізація. Для складніших доменних об'єктів підклас додатково перевизначає методи перетворення, явно вказуючи лише ті поля, що є значущими для конкретної сутності, та окремо керує зв'язками типу «багато до багатьох». Такий підхід дає змогу підкласам точково розширювати базову поведінку без дублювання незмінної логіки валідації та фіксації транзакції.

Кроку валідації відведена особлива роль. Застосовано патерн Strategy у поєднанні зі звітною моделлю (Validation Report): кожне бізнес-правило інкапсульовано в окремому класі, що записує знайдені порушення до спільного звіту без переривання перевірки. Базовий валідатор послідовно виконує всі зареєстровані правила і повертає звіт з повним переліком помилок – на відміну від підходу з генерацією виключення за першої помилки, це дає змогу користувачу отримати всі порушення одночасно.

Правила поділяються на два рівні: узагальнені (перевірка числового діапазону, циклічних посилань у ієрархіях тощо) та доменно-специфічні – складні перевірки, унікальні для конкретної сутності, як-от контроль конфліктів під час складання розкладу, перевірка відповідності кількості годин навантаженню викладача або коректності дат у межах навчального року. Під час видалення порушення цілісності, що виявляються на рівні бази даних, перехоплюються як DbUpdateException і повертаються клієнту у вигляді структурованого звіту.

Висновки. Узагальнений сервісний клас із параметрами типу DTO є ефективним архітектурним рішенням для уніфікації CRUD-операцій у BLL. Базовий клас фіксує незмінний алгоритм, надаючи похідним класам можливість перевизначати окремі кроки за потреби. Модель валідації на основі патерну Strategy забезпечує отримання повного переліку помилок за один прохід та відкритість системи для розширення без модифікації наявного коду.

Список використаних джерел

1. Richards M. Software Architecture Patterns. *O'Reilly Media.* 2015. URL: https://theswissbay.ch/pdf/Books/Computer%20science/O'Reilly/software-architecture-patterns.pdf

2. Bogard J. Vertical Slice Architecture. 2018. URL: https://www.jimmybogard.com/vertical-slice-architecture/

3. Антонов Ю. С., Калько Д. Р., Маруняк А. О. Development of a generalized data model and data access architecture for academic scheduling systems using .net and blazor. *Scientific practice: modern and classical research methods: Collection of scientific papers «ΛΟΓΟΣ» with Proceedings* of the IX International Scientific and Practical Conference (Boston, April 10, 2026). Boston-Vinnytsia: Primedia eLaunch & UKRLOGOS Group LLC, 2026. P. 114–117. DOI: 10.36074/logos-10.04.2026.020.

4. Антонов Ю. Реалізація підсистеми керування розкладом університету на основі змішаної архітектури. *Вісник Хмельницького національного університету. Серія: Технічні науки.* 2025. Т. 357, № 5.1. С. 13–22. DOI: 10.31891/2307-5732-2025-357-1.

5. Antonov Y. LST-Web-api. URL: https://gitlab.com/antonoveducationprojects/codeexamples/cs/webapi/lst-web-api.git

УДК 004.942

Коваль В. С., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

АЛГОРИТМИ СОРТУВАННЯ В МЕТОДАХ ОБЧИСЛЕННЯ: ЕФЕКТИВНІСТЬ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні інформаційні технології пов'язані з обробкою великих обсягів даних, тому важливу роль у програмуванні та методах обчислення відіграють алгоритми сортування. Вони дають змогу впорядковувати інформацію за певними критеріями, що значно пришвидшує пошук, аналіз та обробку даних. Алгоритми сортування використовуються у базах даних, пошукових системах, системах штучного інтелекту, фінансових обчисленнях та багатьох інших галузях інформаційних технологій.

Алгоритм сортування являє собою послідовність дій, спрямованих на впорядкування елементів масиву або іншої структури даних у певному порядку. Основною метою сортування є оптимізація роботи програмних систем та підвищення ефективності виконання обчислювальних задач. Існує велика кількість алгоритмів сортування, які відрізняються принципом роботи, швидкістю виконання та використанням пам'яті.

Одним із найпростіших алгоритмів є сортування бульбашкою (Bubble Sort). Його принцип роботи полягає у послідовному порівнянні сусідніх елементів та їх обміні місцями. Незважаючи на простоту реалізації, цей алгоритм має високу часову складність $O(n^2)$, тому є неефективним для великих обсягів даних [1]. Подібні характеристики мають сортування вибором (Selection Sort) та сортування вставками (Insertion Sort), які використовуються переважно для невеликих масивів або навчальних цілей.

Для роботи з великими наборами даних застосовуються більш ефективні алгоритми, зокрема швидке сортування (Quick Sort) та сортування злиттям (Merge Sort). Quick Sort використовує метод поділу масиву відносно опорного елементу та забезпечує середню часову складність $O(n\ log\ n)$, що робить його одним із найпоширеніших алгоритмів у сучасному програмуванні [2]. Merge Sort ґрунтується на поділі масиву на частини з подальшим їх об'єднанням у відсортований масив. Основною перевагою цього алгоритму є стабільність та гарантована складність $O(n\ log\ n)$, однак він потребує додаткової пам'яті [3].

Ефективність алгоритмів сортування оцінюється за допомогою часової та просторової складності. Часова складність показує кількість операцій, необхід-

них для сортування, а просторова – обсяг додаткової пам’яті, який використовується алгоритмом. Для оцінки складності найчастіше застосовується нотація Big O. Алгоритми зі складністю $O(n^2)$ значно поступаються за швидкістю алгоритмам зі складністю $O(n\ log\ n)$, особливо під час роботи з великими масивами даних.

Практичне застосування алгоритмів сортування є надзвичайно широким. У базах даних вони використовуються для впорядкування записів та оптимізації пошуку інформації. У фінансових системах сортування допомагає аналізувати ринкові дані та виконувати статистичні обчислення. У сфері машинного навчання алгоритми сортування застосовуються для підготовки наборів даних та оптимізації процесів аналізу інформації. Крім того, сучасні мови програмування містять вбудовані функції сортування, які використовують оптимізовані гібридні алгоритми для забезпечення максимальної продуктивності [4].

Головною проблемою під час вибору алгоритму сортування є необхідність пошуку балансу між швидкістю роботи, використанням пам’яті та стабільністю алгоритму. Для невеликих наборів даних можуть використовуватися прості методи сортування, тоді як для великих інформаційних систем застосовуються більш складні та оптимізовані алгоритми.

Отже, алгоритми сортування є важливим складником сучасних методів обчислення та програмування. Від правильного вибору алгоритму залежить продуктивність програмних систем, швидкість обробки інформації та ефективність роботи комп’ютерних технологій. Удосконалення алгоритмів сортування залишається актуальним напрямом розвитку інформатики та інформаційних технологій.

Список використаних джерел

1. Gaffer J. Game Physics: Numerical Integration. Gaffer On Games. URL: https://gafferongames.com/post/integration_basics/ (дата звернення: 10.05.2026).

2. Sarvesh H. Verlet Integration in Game Physics. Medium. URL: https://medium.com/@sarvesh.h/verlet-integration-in-game-physics-4a7ef7557a07 (дата звернення: 10.05.2026).

3. TutsPlus. Physics in Games: Which Integration Method is Best? URL: https://gamedevelopment.tutsplus.com/tutorials/how-to-create-a-custom-2d-physics-engine-the-core-engine--gamedev-7493 (дата звернення: 10.05.2026).

4. Gaffer J. Fix Your Timestep! Gaffer On Games. URL: https://gafferongames.com/post/fix_your_timestep/ (дата звернення: 10.05.2026).

УДК 004.45

Когут М. Я., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Бабаков Р. М., д-р техн. наук, доцент, професор кафедри інформаційних технологій

СИСТЕМА ВІДДАЛЕНОГО АДМІНІСТРУВАННЯ КОМП'ЮТЕРІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

На сучасному етапі розвитку інформаційних технологій доступ до обчислювальних ресурсів має бути постійним і безперебійним. Водночас переважна більшість персональних комп'ютерів розміщена у локальних мережах за NAT-роутерами, що унеможливлює пряме з'єднання з ними без відповідного налаштування мережі або наявності статичної IP-адреси [3].

Аналіз наявних засобів дистанційного керування свідчить про те, що більшість комерційних рішень, зокрема TeamViewer та AnyDesk, побудовані на закритому програмному коді, що унеможливлює проведення незалежної перевірки безпеки та породжує потенційні ризики для конфіденційності даних користувача [1]. До того ж подібні системи мають критичну залежність від власної хмарної інфраструктури розробника: будь-які технічні неполадки на серверах компанії чи обмеження доступу до сервісу позбавляють користувача можливості керувати власними ресурсами. Саме тому застосування відкритих протоколів та автономних модулів сповіщення є необхідною умовою побудови відмовостійких систем адміністрування [4].

Об'єктом дослідження виступають процеси віддаленого керування операційними системами. Метою роботи є створення автономної багатомодульної системи, що забезпечує захищений доступ до комп'ютера з використанням сучасних каналів зв'язку та технологій тунелювання.

У процесі виконання роботи спроєктовано та реалізовано систему мовою програмування Python [3], побудовану на асинхронній архітектурі та представлену такими ключовими компонентами:

– **Ядро системи** (System Core): реалізовано із залученням бібліотеки *psutil* для відстеження стану ресурсів, модуля *pyautogui* для захоплення графічного виводу та стандартного модуля *subprocess* для виконання системних команд в ізольованих потоках виконання.

– **Модуль ідентифікації:** впроваджено механізм авторизації на основі хешованого «Master-Token», що формується криптографічно стійким способом. Модуль реалізує єдину систему управління сесіями через спеціалізований *middleware*.

– **Мережевий адаптер:** забезпечує подолання обмежень NAT шляхом організації реверс-тунелів засобами SSH-клієнта [4]. Застосування протоколу віддаленого перенаправлення портів дає змогу виділити публічну адресу на віддаленому шлюзі, що усуває потребу у статичній IP-адресі.

– **Система сповіщень:** реалізована як мультипотоковий сервіс, що поєднує бібліотеку *aiogram* для роботи з Telegram API [2] та модуль *smtplib* для безпосередньої взаємодії з поштовими серверами через протокол TLS.

– **Вебінтерфейс:** розроблений на основі мікрофреймворку Flask [1] із використанням шаблонізатора *Jinja2,* що забезпечує мінімальне навантаження на хост-комп'ютер.

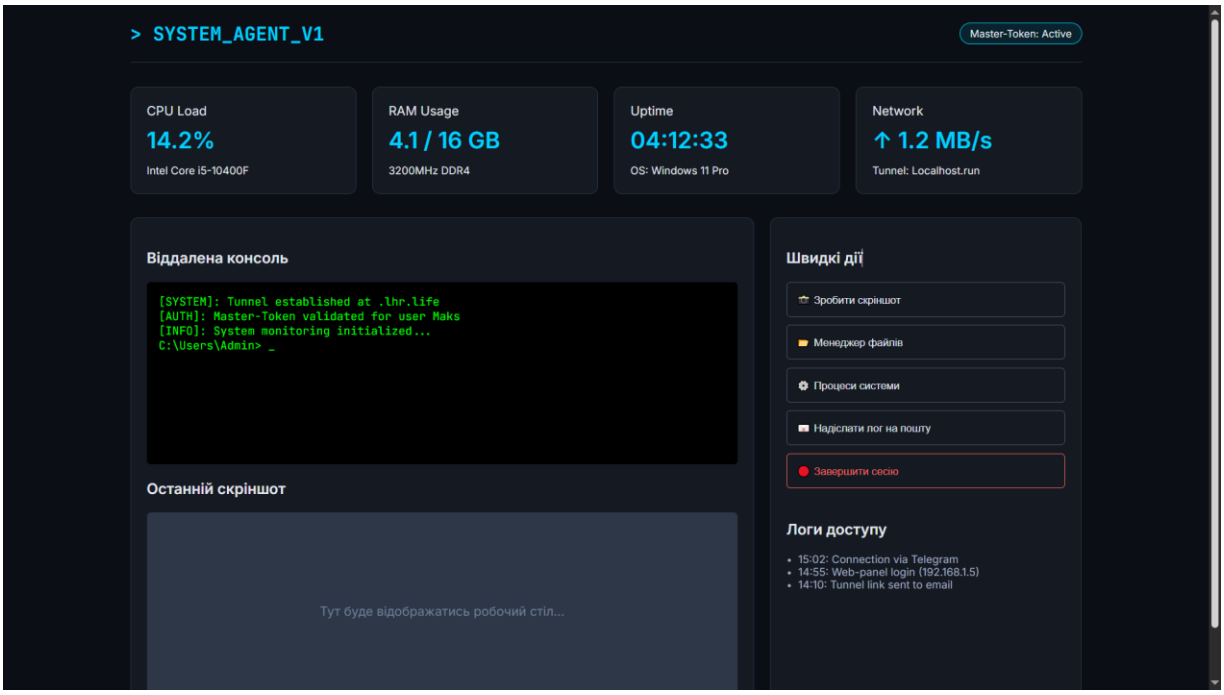


Рисунок 1 – Графічний інтерфейс користувача розробленого агента

Розроблена система продемонструвала високу ефективність у розв'язанні задачі віддаленого доступу до комп'ютерів, розміщених у закритих мережах. Модульна архітектура забезпечує гнучке масштабування функціональних можливостей, а інтеграція одразу двох каналів сповіщення – месенджера та електронної пошти – гарантує стійкість системи до обмежень або збоїв окремих сервісів.

Список використаних джерел

1. Grinberg M. Flask Web Development: Developing Web Applications with Python. 2nd ed. Sebastopol: O'Reilly Media, 2018. 312 p.

2. Aiogram library documentation: official website. 2026. URL: https://docs.aiogram.dev/en/latest/ (дата звернення: 12.04.2026).

3. Python programming language documentation: official website. 2026. URL: https://docs.python.org/3/ (дата звернення: 12.04.2026).

4. Barrett D. J., Silverman R. E., Byrnes R. G. SSH, The Secure Shell: The Definitive Guide. 2nd ed. Sebastopol: O'Reilly Media, 2005. 672 p.

УДК 004.04

Коморний Є. М., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Фриз І. В., канд. фіз.-мат. наук, старший викладач кафедри інформаційних технологій

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ ПІДПРИЄМСТВА

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах цифрової трансформації бізнесу автоматизація внутрішніх процесів підприємства є ключовим фактором підвищення ефективності управління та конкурентоспроможності [1]. Особливо це актуально для малих та середніх підприємств, які потребують доступних програмних рішень для обліку клієнтів, контролю замовлень та формування звітності.

Отже, актуальність теми дослідження зумовлена необхідністю створення ефективних програмних рішень для автоматизації процесів обліку клієнтів та замовлень, що дає змогу мінімізувати вплив людського фактора, зменшити кількість помилок та підвищити продуктивність праці. Незважаючи на наявність великої кількості готових CRM-систем, аналіз наявних рішень показує, що більшість CRM-систем або є надто складними для впровадження, або потребують значних фінансових витрат [2]. Тому існує потреба у гнучких, простих у використанні системах, адаптованих під конкретні бізнес-процеси.

Метою роботи є розробка програмної системи для автоматизації бізнес-процесів підприємства малого та середнього бізнесу. Програмний продукт орієнтований на автоматизацію процесів обліку клієнтів, управління замовленнями, контролю строків виконання та формування аналітичної звітності для малих підприємств.

Архітектура запропонованої системи побудована за модульним принципом та включає рівень інтерфейсу користувача, рівень бізнес-логіки та рівень доступу до даних. Модульний підхід робить код більш організованим, зрозумілим і масштабованим. Замість монолітного коду, де все пов'язане з усім, функціональність програми розділяється на набір незалежних блоків, кожен з яких можна розробляти, тестувати і супроводжувати окремо.

Розробка CRM-системи здійснена з використанням Python, SQLite та бібліотеки CustomTkinter [3]. Використання мови програмування Python обумовлене швидкістю розробки, простотою підтримки та широкими можливостями створення прикладних систем [3; 4], а SQLite обрано завдяки простоті інтеграції та низьким вимогам до ресурсів системи [5]. Для реалізації графічного інтерфейсу використано бібліотеку CustomTkinter, яка забезпечує підтримку сучасного дизайну та стилізацію елементів інтерфейсу.

Для зберігання інформації використовується реляційна база даних SQLite, яка містить таблиці clients, orders та logs. Взаємодія з базою даних реалізована через окремий модуль доступу до даних, який забезпечує виконання SQL-запитів, отримання результатів та централізовану обробку інформації.

Система забезпечує ведення бази клієнтів, створення та обробку замовлень, зміну їх статусів («Нове», «В роботі», «Виконано», «Прострочено»), а також автоматичне визначення прострочених замовлень на основі дати створення або встановленого дедлайну. У системі реалізовано механізм автоматичного визначення прострочених замовлень, пошук клієнтів і замовлень за ключовими параметрами, а також журналювання основних дій користувача.

Особливістю розробленого програмного продукту є реалізація модулів аналітики та звітності, які дають змогу отримувати статистику за періодами, аналізувати кількість замовлень за статусами та оцінювати фінансові показники діяльності підприємства. Передбачено можливість експорту даних у текстові та табличні формати, що дає змогу спростити аналіз діяльності підприємства та підвищити ефективність прийняття управлінських рішень.

Для забезпечення надійності роботи системи реалізовано механізми авторизації користувача, перевірки коректності введених даних та журналювання основних операцій у системі, що дає змогу фіксувати зміни даних і контролювати виконання дій користувача, що підвищує безпеку та цілісність інформації.

Розроблений користувацький інтерфейс має сучасний вигляд та забезпечує зручну взаємодію з системою, включно з пошуком, фільтрацією та переглядом детальної інформації про замовлення. Для підвищення зручності роботи реалізовано механізм автодоповнення під час вибору клієнтів, фільтрацію замовлень та можливість встановлення кінцевих строків виконання.

Отже, розроблена CRM-система забезпечує управління клієнтами та замовленнями, контроль строків виконання, формування звітності та журналювання дій користувача. Запропоноване рішення дає змогу підвищити ефективність управління, зменшити кількість помилок і може слугувати основою для подальшого розширення функціоналу системи.

Список використаних джерел

1. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner’s Approach. 9th ed. New York: McGraw-Hill, 2020.
2. Sharda R., Delen D., Turban E. Business Intelligence, Analytics, and Data Science. 5th ed. Pearson, 2023.
3. Matthes E. Python Crash Course. 3rd ed. San Francisco: No Starch Press, 2023.
4. Python Documentation. URL: https://docs.python.org/3/ (дата звернення: 09.05.2026).
5. SQLite Documentation. URL: https://www.sqlite.org/docs.html (дата звернення: 09.05.2026).

УДК 004.415.2:004.738.5:658.89

Мисько Б. В., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

МЕТОДИ ПРОЄКТУВАННЯ МОДУЛЬНОЇ АРХІТЕКТУРИ СЕРВЕРНОЇ ЧАСТИНИ CRM-СИСТЕМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах цифровізації освітніх послуг CRM-системи набувають важливого значення для організації роботи навчальних закладів, зокрема шкіл іноземних мов. Такі системи дають змогу автоматизувати облік клієнтів, управління заявками, розкладом занять, оплатами, групами, викладачами та комунікацією з учнями. Однак ефективність CRM-системи значною мірою залежить не лише від її функціональних можливостей, а й від правильно спроєктованої серверної частини.

Серверна частина веборієнтованої CRM-системи відповідає за обробку запитів користувачів, взаємодію з базою даних, реалізацію бізнес-логіки, контроль доступу, збереження даних та передачу інформації між клієнтською частиною і внутрішніми модулями системи. У разі відсутності чіткої архітектури така система швидко стає складною для підтримки, розширення та тестування. Особливо це актуально для CRM-систем, які з часом можуть доповнюватися новими функціями.

Метою дослідження є визначення основних методів проєктування модульної архітектури серверної частини CRM-системи для школи іноземних мов. У межах роботи розглядаються підходи до поділу серверної частини на окремі функціональні модулі, організації взаємодії між ними, розмежування відповідальності компонентів та забезпечення можливості подальшого розвитку системи.

Модульна архітектура передбачає поділ програмної системи на окремі логічні частини, кожна з яких відповідає за певну групу функцій. Для CRM-системи школи іноземних мов доцільно виділити такі основні модулі: користувачі, автентифікація та авторизація, клієнти, навчальні групи, розклад, заявки, оплати та звітність. Такий підхід дає змогу уникнути надмірного змішування логіки, коли різні функції системи реалізуються в одному місці без чіткого поділу.

Одним із ключових методів проєктування модульної архітектури є розмежування відповідальності між рівнями серверної частини. На рівні контролерів доцільно обробляти вхідні HTTP-запити та формувати відповіді для клієнтської частини. На рівні сервісів має розміщуватися основна бізнес-логіка системи. Рівень роботи з даними повинен відповідати за звернення до бази даних, виконання запитів та повернення результатів у зручному для обробки вигляді. Такий поділ робить систему більш зрозумілою, оскільки кожен компонент має конкретне призначення.

Важливим методом є проєктування модулів за предметними областями. У CRM-системі для школи іноземних мов предметними областями можуть бути учні, викладачі, групи, курси, розклад, оплати та комунікації. Наприклад, модуль розкладу не повинен напряму відповідати за фінансові операції, а модуль оплат не повинен містити логіку формування навчальних груп. Завдяки цьому кожна частина системи залишається відносно самостійною, а зміни в одному модулі менше впливають на інші.

Окрему роль у модульній архітектурі відіграє метод використання єдиного інтерфейсу взаємодії між клієнтською та серверною частинами. Найчастіше для цього застосовується REST API, яке забезпечує передачу даних через стандартизовані HTTP-запити. Для CRM-системи це дає змогу створити зрозумілу структуру маршрутів для роботи з користувачами, групами, заявками, оплатами та іншими сутностями. До того ж API дає змогу в майбутньому підключати до системи не лише вебінтерфейс, а й мобільний застосунок або зовнішні сервіси.

Для реалізації серверної частини доцільним є використання Node.js, оскільки ця платформа орієнтована на асинхронну обробку запитів і добре підходить для вебзастосунків, які активно взаємодіють з базою даних та зовнішніми сервісами. Механізм подій у Node.js дає змогу виконувати неблокуючі операції введення-виведення, що є важливим для систем, де одночасно працюють адміністратори, менеджери, викладачі та учні [1]. Офіційна документація Node.js пояснює, що event loop забезпечує неблокуюче виконання операцій введення-виведення, незважаючи на використання одного основного потоку JavaScript.

Практичним способом організації модульної серверної частини є використання фреймворків Express.js або NestJS. Express.js надає гнучкі можливості для побудови маршрутів і middleware, що дає змогу розділяти обробку запитів, перевірку доступу, валідацію даних та обробку помилок [2]. Натомість NestJS насамперед орієнтований на модульну структуру застосунку, де кожен модуль може містити власні контролери, сервіси та провайдери [3]. В офіційній документації NestJS модуль розглядається як основний елемент організації структури застосунку, а провайдери використовуються для побудови взаємозв’язків між сервісами та іншими компонентами.

До важливих методів проєктування також належить використання принципу слабкої зв’язаності між модулями. Це означає, що модулі не повинні напряму залежати від внутрішньої реалізації один одного. Наприклад, модуль заявок може отримувати інформацію про клієнта через сервіс клієнтів, але не повинен самостійно змінювати структуру даних цього модуля. Такий підхід спрощує тестування, зменшує кількість помилок під час внесення змін і дає змогу поступово розширювати систему.

Ще одним важливим методом є централізована обробка помилок та валідація вхідних даних. У CRM-системі обробляється велика кількість інформації, тому якщо перевірка даних буде реалізована хаотично, це може призвести до збереження некоректної інформації або порушення логіки роботи системи. Тому доцільно передбачити єдині правила валідації для всіх модулів та спільний механізм повернення помилок клієнтській частині.

Модульна архітектура також спрощує подальше масштабування CRM-системи. На початковому етапі система може бути реалізована як модульний моноліт, тобто як єдиний серверний застосунок із чітким внутрішнім поділом на модулі. Такий підхід є зручним для невеликої або середньої команди розробників, оскільки не потребує складної інфраструктури, але водночас зберігає можливість подальшого переходу до мікросервісної архітектури. У практиці розробки програмного забезпечення модульний моноліт часто розглядається як більш контрольований початковий етап перед можливим виділенням окремих сервісів [4]. Підхід «спочатку моноліт» також підтримується в архітектурних рекомендаціях, де наголошується, що нову систему доцільно починати з простішої структури, якщо немає достатніх підстав одразу переходити до мікросервісів. Для CRM-системи школи іноземних мов такий підхід є особливо доцільним, оскільки система має багато взаємопов'язаних процесів, які не мають бути змішані в одному програмному блоці. Саме тому модульна архітектура дає змогу зберегти логічну цілісність системи та водночас забезпечити зрозумілу структуру її серверної частини.

Отже, проєктування модульної архітектури серверної частини CRM-системи є важливим етапом створення надійного веборієнтованого програмного забезпечення. Основними методами такого проєктування є поділ системи на функціональні модулі, розмежування відповідальності між рівнями, використання REST API, впровадження централізованої обробки помилок, валідації даних та забезпечення слабкої зв'язаності між компонентами. Застосування цих підходів дає змогу структурувати систему відповідно до її основних бізнес-процесів, зменшити залежність між окремими частинами програмного забезпечення, спростити подальше додавання нових функцій та створити серверну частину, зручну для підтримки й розширення. Використання Node.js у поєднанні з REST API та модульною організацією коду створює основу для побудови гнучкої, масштабованої та зручної в супроводі CRM-системи, яка враховує як технічні вимоги до сучасного вебзастосунку, так і практичні потреби освітньої установи: облік клієнтів, управління групами, розкладом, заявками, оплатами та ролями користувачів.

Список використаних джерел

1. Node.js Documentation. The Node.js Event Loop. URL: https://nodejs.org/learn/asynchronous-work/event-loop-timers-and-nexttick

2. Express.js Documentation. Using middleware; Routing. URL: https://expressjs.com/en/guide/writing-middleware.html

3. NestJS Documentation. Modules; Providers. URL: https://docs.nestjs.com/modules

4. Fowler M. Monolith First. *MartinFowler.com.* URL: https://martinfowler.com/bliki/MonolithFirst.html

УДК 004.738.1

Підруцький Д. А., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

АРХІТЕКТУРНІ ОСОБЛИВОСТІ ПОБУДОВИ МАСШТАБОВАНИХ МАРКЕТПЛЕЙСІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасній веброзробці особливої актуальності набуває побудова складних систем електронної комерції – маркетплейсів, які вимагають високої продуктивності, надійності та здатності до швидкого масштабування [1]. Одним із найпоширеніших інструментів для створення серверної частини таких проєктів є середовище Node.js, а для реалізації гнучкої логіки обробки запитів активно використовується фреймворк Express [3].

Актуальність теми зумовлена потребою у створенні відмовостійких архітектур, здатних ефективно обробляти велику кількість паралельних з'єднань та забезпечувати цілісність даних у контексті сучасних стандартів високонавантажених систем.

Метою роботи є розробка архітектури серверної частини маркетплейса з використанням стека PERN для демонстрації ефективності модульного підходу та сучасних інструментів управління даними у побудові масштабованих вебресурсів.

У процесі роботи розроблено архітектурну модель вебзастосунку, що включає рівні обробки запитів, бізнес-логіки та взаємодії з базою даних. За допомогою Prisma ORM реалізовано типізований доступ до PostgreSQL, що забезпечило стабільність роботи зі складними структурами даних [2]. Також використано хмарний сервіс Clerk для делегування завдань автентифікації, що дало змогу знизити навантаження на основний сервер та підвищити безпеку системи.

Перевагами такого підходу є чітке розділення відповідальності між компонентами, висока швидкість розробки та можливість легкого впровадження нових функцій, як-от системи предиктивної аналітики чи рекомендацій. Використання Node.js у поєднанні з сучасними ORM-рішеннями є особливо ефективним у розробці великих торговельних майданчиків, де критично важливою є швидкість відгуку API [3].

Отже, побудова маркетплейса на основі Node.js, Express та Prisma є сучасним та ефективним підходом до розробки вебзастосунків, який забезпечує продуктивність, масштабованість і зручність подальшого супроводу системи.

Список використаних джерел

1. Що таке масштабованість: 10 ключових понять. URL: https://it-rating.ua/shho-take-masshtabovanist-veb-dodatkiv (дата звернення: 07.05.2026).

2. Prisma Documentation: Data modeling and ORM for modern applications. URL: https://www.prisma.io/docs (дата звернення: 07.05.2026).
3. 5 просунутих прийомів NodeJS з ExpressJS. DevZone. URL: https://devzone.org.ua/post/5-prosunutix-priiomiv-nodejs-z-expressjs (дата звернення: 07.05.2026).

УДК 004.42:004.738.5

Поліщук О. С., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

РОЗРОБКА АДАПТИВНОЇ ВЕБПЛАТФОРМИ ПОШУКУ ТА ФІЛЬТРАЦІЇ НЕРУХОМОСТІ НА REACT

Донецький національний університет імені Василя Стуса, м. Вінниця

Актуальність дослідження. Сучасний ринок нерухомості перебуває на етапі активної цифрової трансформації, що супроводжується зростанням ролі онлайн-платформ у процесах пошуку, аналізу та придбання житла. Використання вебтехнологій дає змогу користувачам отримувати швидкий доступ до великого обсягу інформації, здійснювати порівняння об'єктів і приймати обґрунтовані рішення [1].

Згідно з аналітичними дослідженнями, частка онлайн-взаємодії на ринку нерухомості постійно зростає, що пояснюється зручністю використання цифрових сервісів і можливістю застосування багатопараметричних фільтрів [2]. Водночас наявні платформи мають низку недоліків, серед яких складність інтерфейсу, недостатня швидкість роботи, обмежена персоналізація та не завжди актуальні дані [3].

Особливої актуальності набуває проблема адаптивності інтерфейсу, оскільки значна частина користувачів здійснює пошук нерухомості з мобільних пристроїв. Це вимагає створення вебплатформ, які забезпечують зручність і стабільність роботи на різних типах пристроїв [4].

Мета та завдання роботи. Метою роботи є розробка адаптивної вебплатформи пошуку та фільтрації нерухомості **EstatePro,** яка забезпечує ефективну взаємодію користувачів із системою.

Основні завдання:

- аналіз сучасного стану ринку онлайн-сервісів нерухомості;
- дослідження функціональних можливостей наявних платформ;
- формування вимог до системи;
- проєктування архітектури;
- розробка алгоритмів пошуку та фільтрації;
- реалізація програмного рішення;
- тестування та оцінка ефективності.

Аналіз наявних рішень. Сучасні вебплатформи пошуку нерухомості використовують багатокритеріальну фільтрацію, що дає змогу враховувати параметри, як-от ціна, місцезнаходження, тип об’єкта та інші характеристики [1]. Значну роль відіграє інтеграція з геолокаційними сервісами, які забезпечують візуалізацію об’єктів на карті.

Окремі платформи застосовують елементи штучного інтелекту для формування персоналізованих рекомендацій, що підвищує ефективність пошуку [2]. Проте існують обмеження, пов’язані з якістю даних, складністю інтерфейсів та недостатньою адаптивністю систем [3].

Архітектура та технології. Для реалізації вебплатформи було обрано клієнт-серверну архітектуру, яка забезпечує розділення логіки між інтерфейсом користувача та серверною частиною.

Клієнтська частина реалізована як односторінковий додаток із використанням React, що дає змогу динамічно оновлювати контент без перезавантаження сторінки та забезпечує високий рівень зручності користування [4].

Серверна частина побудована на Node.js та Express, що дає змогу ефективно обробляти запити та реалізувати взаємодію з базою даних через API [5].

Для зберігання даних використано SQLite, яка забезпечує надійність і достатню продуктивність для даного типу систем [6].

Реалізація функціоналу. Розроблена вебплатформа включає такі функціональні можливості:

– пошук і фільтрація об’єктів нерухомості;
– перегляд детальної інформації;
– система обраного та історії переглядів;
– особисті кабінети користувачів;
– модерація оголошень і управління ролями.

Інтерфейс реалізовано на основі компонентного підходу, що забезпечує гнучкість і масштабованість системи [4].

Алгоритми пошуку та безпека. У системі реалізовано комбінований підхід до пошуку: серверна обробка запитів за допомогою SQL та клієнтська фільтрація для уточнення результатів. Це дає змогу забезпечити швидку роботу та зручність взаємодії з користувачем [5].

Для забезпечення безпеки використано JWT-автентифікацію, що дає змогу захистити доступ до ресурсів і персональних даних користувачів [7].

Висновки. За результатами виконання роботи було розроблено адаптивну вебплатформу пошуку нерухомості, яка відповідає сучасним вимогам до вебдодатків. Використання сучасного технологічного стеку дало змогу створити ефективний інструмент для роботи з нерухомістю.

Отримані результати підтверджують доцільність використання клієнт-серверної архітектури та сучасних вебтехнологій. Розроблена система може бути використана як основа для подальшого розвитку комерційних сервісів.

Список використаних джерел

1. Statista. Real Estate Market Analysis. URL: https://www.statista.com
2. Zillow Research. Housing Market Trends. URL: https://www.zillow.com/research

3. LUN.ua. Real Estate Market Analytics. URL: https://lun.ua
4. React Documentation. URL: https://react.dev
5. Node.js Documentation. URL: https://nodejs.org
6. SQLite Documentation. URL: https://www.sqlite.org/docs.html
7. JWT. JSON Web Token Introduction. URL: https://jwt.io/introduction

УДК 004.928

Попова К. О., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Фриз І. В., канд. фіз.-мат. наук, старший викладач кафедри інформаційних технологій

РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ У ЖАНРІ ГОЛОВОЛОМКИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Індустрія комп'ютерних ігор розвивається стрімкими темпами, але особливе місце в ній займають комп'ютерні ігри-головоломки. Класичні головоломки, як-от Sokoban, мають високу цінність, але їх сталий ігровий процес часто є занадто статичним для сучасного гравця.

Актуальність цього дослідження полягає у модернізації традиційної покрокової механіки шляхом інтеграції динамічних елементів, в тому числі обмеження часу, що стимулює швидке прийняття рішень та підвищує загальну залученість користувача.

Дослідження наяваних ігор-аналогів Sokoban, як-от MazezaM, Sokowand та Vextorial, показало, що вони мають дуже багато ігрових елементів та механік, які є або сильно статичними, або навпаки – перезавантаженими і навіть зайвими. Більшість із них не пропонують гравцеві часових обмежень, що знижує динаміку ігрового процесу, або не мають інструментів швидкого перезавантаження сцени в разі логічного тупика [1].

Метою дослідження є проєктування та програмна реалізація двовимірної комп'ютерної гри-головоломки, яка поєднує в собі класичні правила штовхання блоків із зворотним відліком часу.

Для досягнення цієї мети необхідно вирішити такі **завдання:** обґрунтувати технологічний стек розробки, спроєктувати архітектуру взаємодії ігрових компонентів та реалізувати надійні алгоритми дискретного руху й перевірки колізій.

Для реалізації проєкту обрано інструментарій, що складається з ігрового рушія Unity, мови програмування C# та інтегрованого середовища розробки Microsoft Visual Studio [2–5]. Unity забезпечує високу оптимізацію під 2D-проєкти та має вбудовану компонентну систему фізики Physics2D. Вибір мови C# зумовлений її типізацією, що виключає помилки у прорахунках координат сітки, також має автоматичне керування пам'яттю, що гарантує стабільну частоту кадрів. Середовище Visual Studio забезпечує глибоку інтеграцію з Unity та надає потужні інструменти покрокового налагодження коду.

Архітектура програми побудована за компонентно-орієнтованим принципом. Взаємодія між об’єктами на сцені керується чітким алгоритмом. Загальну логіку функціонування та перевірки умов ходу персонажа представлено на рис. 1.

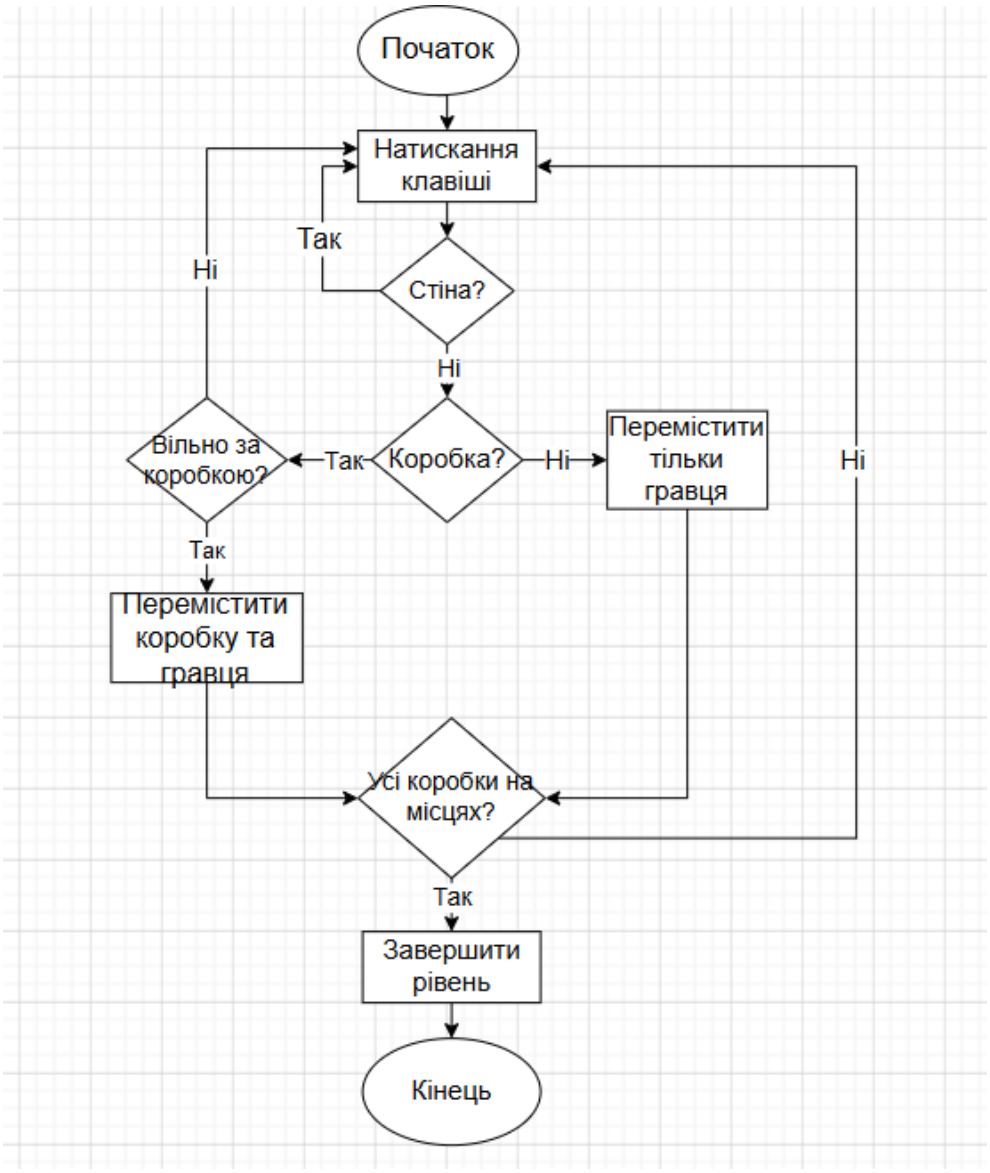


Рисунок 1 – Блок-схема алгоритму ігрового процесу

Центральним елементом механіки є дискретне переміщення гравця за допомогою клавіатурного введення. Для запобігання помилок під час накладання об’єктів у скрипті MovementPlayer.cs реалізовано метод TryMove, який за допомогою функції Physics2D.OverlapCircle сканує простір на один крок вперед: якщо попереду виявлено шар стіни, рух блокується; якщо виявлено шар коробки, алгоритм прораховує вільне місце за нею; коробка і гравець зміщуються лише тоді, коли наступна клітинка за коробкою є абсолютно порожньою.

Для відстеження стану гри та прогресу розроблено класи TargetCheck та GameManager. Скрипт TargetCheck.cs встановлено на цільові зони-цілі. За допомогою системних методів OnTriggerEnter2D та OnTriggerExit2D він фіксує появу або зникнення коробки в зоні, змінює логічний прапорець isOccupied та пере-

дає сигнал до GameManager. Клас GameManager здійснює централізований підрахунок: під час досягнення ліміту активованих зон він зупиняє гру та активує графічне вікно перемоги. Одночасно з цим у методі Update функціонує таймер зворотного відліку. Якщо час вичерпується раніше, ніж гравець розставив усі блоки, GameManager фіксує програш.

Для зручності користувача реалізовано допоміжні класи: MainMenu (керування кнопками стартового меню), NextLevel (завантаження наступної сцени через Build Index) та RestartLevel, який після натискання гарячої клавіші «R» миттєво перезавантажує поточну сцену, даючи змогу вийти з логічного глухого кута.

За результатами роботи було успішно спроєктовано та розроблено прототип 2D-комп'ютерної гри-головоломки на рушії Unity. Створені алгоритми забезпечують точну дискретну взаємодію об'єктів без збоїв у системі колізій. Програмний продукт повністю відповідає висунутим функціональним вимогам, є оптимізованим та готовим до подальшого розвитку.

Список використаних джерел

1. Top similar games like Sokoban (Boxman) Classic. URL: https://steampeek.hu/?from=36&appid=1406570 (дата звернення 08.05.2026).
2. Що таке Unity? URL: https://www.tecnoloblog.com/uk/ (дата звернення 08.05.2026).
3. Ward T. Using Unity to Help Solve Intelligence. 2020. URL: https://arxiv.org/abs/2011.09294 (дата звернення 08.05.2026).
4. Goldstone W. Unity Game Development Essentials. Packt Publishing, 2013. 316 p.
5. Price M. C# 10 and .NET 6 – Modern Cross-Platform Development. Packt Publishing, 2021. 826 p.

УДК 004.9:004.421.2

Приходський Г. С., здобувач вищої освіти
4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент,
доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ РУШІЯ GODOT ДЛЯ РОЗРОБКИ 2D-ІГОР

Донецький національний університет імені Василя Стуса, м. Вінниця

Ігрова індустрія є однією з найдинамічніших сфер сучасних цифрових технологій, яка активно розвивається протягом останніх десятиліть. Від простих аркадних автоматів до складних інтерактивних світів відеоігри пройшли значний шлях еволюції. Вони стали не лише засобами розваги, а й важливим складником культури, освіти й мистецтва. Особливо помітним є вплив відеоігор у країнах Азії, де індустрія має потужну підтримку та відзначається унікальними підходами до створення ігрового процесу та візуального стилю. Це підтверджує глобальний характер розвитку ігрової розробки та її значення у сучасному суспільстві [1].

Одним із ключових аспектів створення успішної гри є її дизайн та атмосфера. Дизайн гри визначає характер взаємодії гравця з ігровим середовищем, емоції, які він відчуває, та рівень залученості до процесу. Як зазначає Джессі Шелл, гру доцільно розглядати як систему взаємопов'язаних елементів, де кожен компонент впливає на загальний досвід користувача. Важливим складником діяльності розробника є аналіз поведінки гравця та здатність враховувати різні аспекти взаємодії з грою, що дає змогу створювати цілісний ігровий досвід [2].

Важливу роль у дизайні гри відіграють ігрові механіки – правила та способи взаємодії гравця з ігровим середовищем. До них належать рух персонажа, взаємодія з об'єктами, бонуси, вороги та інші елементи, які формують ігровий процес. Згідно з підходом Шелла, гра може розглядатися як формалізована система з правилами, цілями та викликами, що передбачає активну участь гравця у вирішенні ігрових завдань. Цей підхід дає змогу трактувати ігровий процес як процес прийняття рішень та подолання труднощів [2].

Сучасні ігрові рушії значно спрощують процес розробки ігор, надаючи готові інструменти для реалізації графіки, фізики, анімації та програмної логіки. Одним із таких рушіїв є Godot, який є безкоштовним програмним забезпеченням із відкритим вихідним кодом. Він підтримує створення як 2D-, так і 3D-ігор, однак особливо зручним є для розробки 2D-проєктів завдяки спеціалізованим інструментам. Основною особливістю Godot є сцено-орієнтована архітектура, у межах якої кожен елемент гри представлений у вигляді вузла, а сцени можуть комбінуватися між собою, формуючи складну ієрархічну структуру [3; 4].

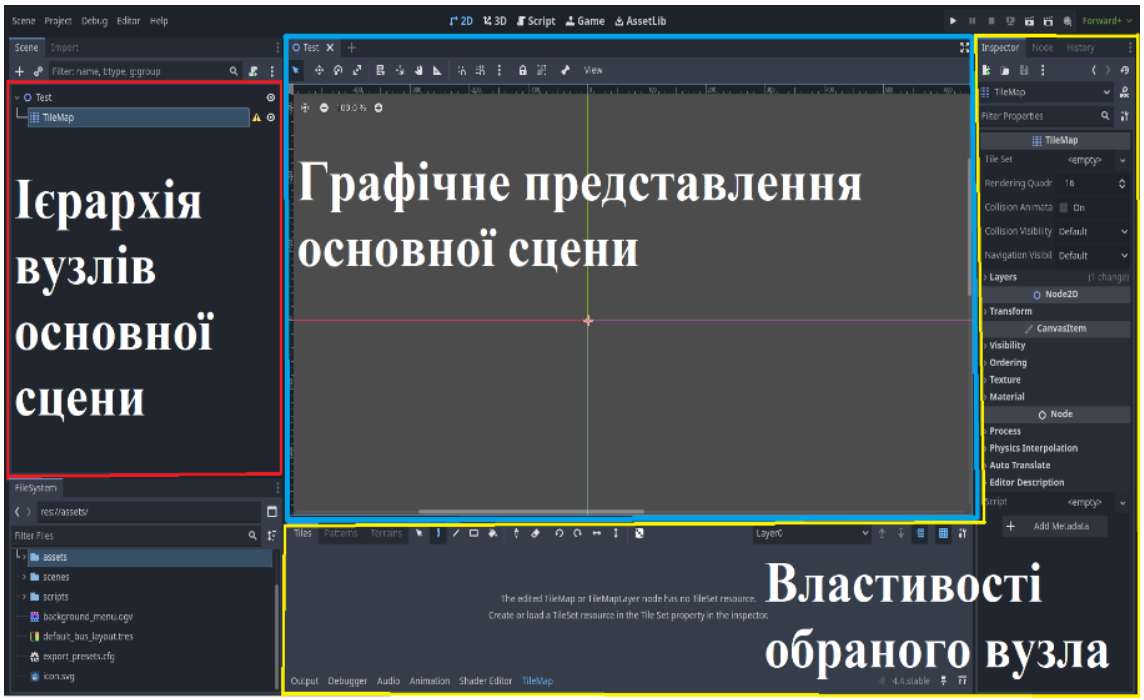


Рисунок 1 – Інтерфейс середовища розробки у Godot

Згідно з документацією Godot, важливим аспектом є модульний підхід до розробки, що дає змогу створювати незалежні компоненти та повторно використовувати їх у межах проєкту. Такий підхід забезпечує гнучкість структури

програми та спрощує підтримку коду, оскільки елементи ігрової логіки можуть розроблятися та тестуватися незалежно один від одного. До того ж рушій має власну мову програмування GDScript, синтаксис якої подібний до мови Python, що значно спрощує реалізацію ігрової логіки та прискорює процес розробки [4]. Реалізацію ігрового рівня у жанрі 2D-платформер наведено на рис. 2.

Практичне застосування рушія Godot розглянуто у праці Кріса Бредфілда, де демонструється покрокове створення ігрових проєктів різних жанрів. У межах розробки 2D-гри описано процес створення головного персонажа, налаштування його руху, анімацій і системи зіткнень, а також реалізацію інтерактивних об’єктів, зокрема предметів для збору та елементів інтерфейсу. Окрему увагу приділено організації сцен, написанню скриптів та побудові користувацького інтерфейсу, що формує основу ігрового процесу [3].

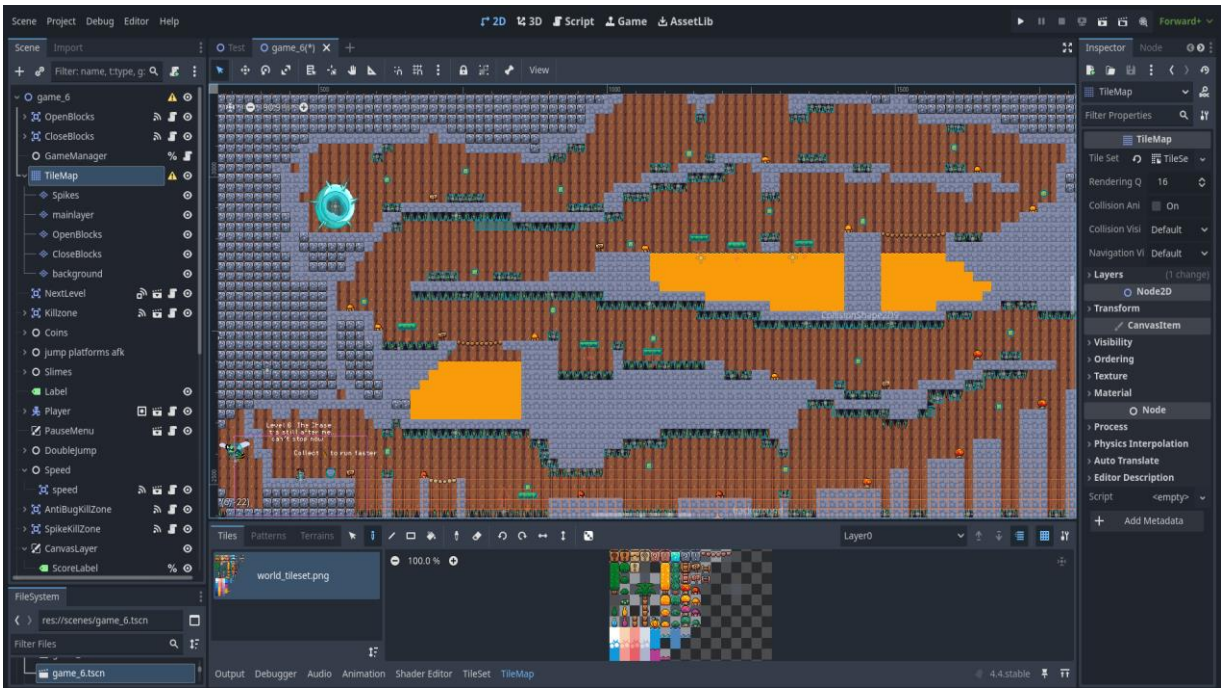


Рисунок 2 – Приклад реалізованого ігрового рівня у жанрі 2D-платформер

Особлива увага приділяється створенню ігрового середовища та рівнів. Використання тайлів і вузла TileMap дає змогу ефективно будувати структуру рівнів, а також реалізовувати прокрутку фону та розміщення елементів. У процесі розробки платформера розглядається додавання ворогів, налаштування їх поведінки, створення системи бонусів і поступове ускладнення ігрового процесу. Це забезпечує динамічність гри та підтримує інтерес гравця [3].

Отже, використання рушія Godot для розробки 2D-ігор дає змогу ефективно реалізувати всі основні компоненти сучасного ігрового продукту – від проєктування дизайну до програмної реалізації. Поєднання теоретичних підходів до створення ігор та практичних інструментів розробки створює основу для розробки якісних ігрових продуктів. Godot є потужним інструментом, який відкриває широкі можливості як для початківців, так і для досвідчених розробників.

Список використаних джерел

1. Cario E. Press Start: The History of Video Gaming. Ilex Press, 2025. 288 p.
2. Schell J. The Art of Game Design: A Book of Lenses. CRC Press, 2019. 652 p.
3. Bradfield C. Godot 4 Game Development Projects: Build five cross-platform 2D and 3D games using one of the most powerful open-source game engines. Packt Publishing, 2023. 264 h.
4. Документація Godot Engine українською мовою. URL: https://docs.godotengine.org/uk/4.x/index.html

УДК 004.43:004.773.3

Труханська В. О., здобувачка вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

АНАЛІЗ СУЧАСНИХ JAVASCRIPT-ФРЕЙМВОРКІВ ДЛЯ ПОБУДОВИ КЛІЄНТСЬКОЇ ЧАСТИНИ ВЕБОРІЄНТОВАНОЇ CRM-СИСТЕМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Цифровізація освіти зумовлює підвищення попиту на спеціалізовані інформаційні системи управління відносинами з клієнтами для закладів освіти. Школи іноземних мов є типовими представниками ринку приватного навчання, де якісне управління студентами, розкладом, платежами і комунікаціями безпосередньо впливає на конкурентоспроможність закладу. Клієнтська частина такої системи виступає основним інтерфейсом взаємодії персоналу з даними, тому архітектурні рішення на її рівні визначають функціональну надійність, швидкодію та масштабованість усього програмного комплексу [1].

Вибір JavaScript-фреймворку є ключовим рішенням під час розробки сучасних веборієнтованих застосунків. Від нього залежать продуктивність рендерингу інтерфейсу, масштабованість компонентної архітектури, швидкість розробки та довгострокова підтримуваність коду. Метою цієї роботи є проведення порівняльного аналізу провідних JavaScript-фреймворків та обґрунтування вибору оптимального технологічного стеку для клієнтської частини CRM-системи школи іноземних мов.

Станом на 2024–2025 рр. у сфері front-end-розробки домінують чотири JavaScript-фреймворки: React від компанії Meta, Vue.js, Angular від Google та Svelte. Їх сумарна частка у щотижневих завантаженнях npm-реєстру перевищує 95 % від загального обсягу завантажень інструментів для побудови веборієнтованих застосунків. React використовує декларативний підхід на основі Virtual DOM з однонаправленим потоком даних та функціональною моделлю управління станом. Vue.js поєднує систему відстеження змін даних із Composition API та гнучкою структурою компонентів. Angular є повноцінним MVC-фреймворком зі вбудованим контейнером впровадження залежностей та строгою

типізацією. Svelte компілює компоненти у нативний JavaScript без Virtual DOM, що забезпечує мінімальний розмір генерованого коду [2].

Для об'єктивного порівняння фреймворків визначено три групи критеріїв. Технічні критерії охоплюють розмір фінального коду застосунку, швидкість рендерингу інтерфейсу та підтримку TypeScript. Архітектурні критерії включають підхід до управління глобальним станом застосунку, наявність зрілих UI-бібліотек для бізнес-компонентів та загальну масштабованість у разі зростання кількості модулів. Організаційні критерії враховують складність первинного освоєння технології, активність підтримки спільноти та доступність спеціалістів на ринку праці [2].

Таблиця 1 – Порівняльна характеристика JavaScript-фреймворків

Критерій	**React**	**Vue.js**	**Angular**	**Svelte**
Розмір bundle (gzip, КБ)	~42	~34	~78	~10
Час рендерингу (мс)	2,1	1,9	3,4	1,5
Складність освоєння	Низька	Низька	Висока	Низька
Підтримка TypeScript	Висока	Висока	Нативна	Висока
Стейт-менеджмент	Redux Toolkit	Pinia	NgRx	Stores
UI-бібліотеки	MUI, Ant Design	Vuetify	Angular Material	Обмежено
Завантаження npm/тиж.	~25 млн	~5 млн	~4 млн	~1 млн
Екосистема (1–5)	5	4	4	3

Аналіз наведених даних у табл. 1 виявляє суттєві відмінності між фреймворками. Angular вирізняється найвищим рівнем архітектурної структурованості, проте має найбільший розмір bundle – приблизно 78 Кб – та значно вищий поріг освоєння, що обмежує його доцільність для проєктів середнього масштабу [3]. Svelte генерує найменший розмір коду завдяки компіляційному підходу, однак поступається за зрілістю екосистеми бізнес-компонентів, що є критичним під час розробки CRM-інтерфейсів зі складними таблицями та формами. Vue.js демонструє збалансовані показники, але має у п'ять разів менший обсяг тижневих завантажень, порівняно з React, що відображається у вужчому ринку розробників. React із розміром bundle приблизно 42 Кб і часом рендерингу 2,1 мс забезпечує оптимальне поєднання продуктивності та зрілості екосистеми [4].

CRM-система школи іноземних мов включає модулі управління студентами та групами, інтерактивний розклад занять, журнал відвідуваності та успішності, блок фінансових операцій і аналітичну панель. Компонентна архітектура React забезпечує природну декомпозицію складного інтерфейсу на ізольовані блоки, що повторно використовуються в різних частинах системи. Redux Toolkit забезпечує передбачуваний потік даних за одночасної роботи з множинними джерелами – студентами, розкладами та платежами. Для роботи з формами реєстрації й оплат застосовується React Hook Form у поєднанні з Zod для декларативної валідації даних, а React Query забезпечує оптимізований кешований обмін даними з REST API-сервера [4]. Готові бібліотеки компонентів MUI та Ant Design містять весь необхідний набір елементів для бізнес-застосунків – таблиці, модальні вікна, форми, фільтри та графіки, що суттєво скорочує час розробки.

Наукова новизна роботи полягає у розробці системи критеріїв порівняльного аналізу, адаптованої до специфіки веборієнтованих CRM-систем освітніх закладів. На відміну від наявних оглядових досліджень, запропонована методика враховує функціональний профіль конкретного класу систем і може бути відтворена під час розробки аналогічних рішень. Практична цінність полягає у безпосередньому застосуванні обґрунтованих рішень під час розробки клієнтської частини CRM-системи школи іноземних мов у межах магістерської роботи.

Проведений порівняльний аналіз підтверджує, що React є оптимальним вибором для побудови клієнтської частини CRM-системи школи іноземних мов завдяки збалансованому рівню продуктивності, найбільш розвиненій екосистемі готових бізнес-компонентів, зрілим інструментам управління станом та найвищій доступності спеціалістів серед усіх розглянутих альтернатив. Запропонована система критеріїв є відтворюваною і може застосовуватись під час вибору технологічного стеку для інших класів веборієнтованих бізнес-систем. Перспективою подальших досліджень є порівняльне оцінювання продуктивності реалізованої CRM-системи на базі React із застосуванням серверного рендерингу на основі Next.js.

Список використаних джерел

1. Vulić M., Barać D., Bogdanović Z., CRM as a cloud service in e-education. *19th Telecommunications Forum (TELFOR) Proceedings of Papers* (Belgrade, Serbia, 2011). 2011. P. 1470–1473. DOI: 10.1109/TELFOR.2011.6143834 (дата звернення: 06.05.2025).

2. Emmanni Ph. S. Comparative Analysis of Angular, React, and Vue.js in Single Page Application Development. *International Journal of Science and Research (IJSR).* 2023. Vol. 12. DOI: 10.21275/SR24401230015. (дата звернення: 06.05.2025).

3. Interpretation and Analysis of Angular Framework / G. Geetha, M. Mittal, K. M. Prasad, G. Ponsam. *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS). IEEE.* 2022. DOI: 10.1109/ICPECTS56089.2022.10047474 (дата звернення: 06.05.2025).

4. Maratkar P. S., Adkar P. React JS – An Emerging Frontend JavaScript Library. *Iconic Research and Engineering Journals.* 2021. Vol. 4, № 12. P. 99–102 (дата звернення: 06.05.2025).

УДК 004.3

Тупіцький В. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ПОВЕДІНКОВІ ЕКОНОМІЧНІ ПАТЕРНИ В РОЗРОБКАХ КОМП'ЮТЕРНИХ СТРАТЕГІЧНИХ ІГОР

Донецький національний університет імені Василя Стуса, м. Вінниця

Стратегічні відеоігри давно вийшли за межі простого розважання. Сучасні ігри жанру реального часу (RTS) та покрокові стратегії (TBS) пропонують гли-

бокі економічні симуляції, що моделюють як мікроекономічні рішення окремих агентів, так і макроекономічні процеси на рівні держави або цивілізації [1]. Розуміння цих механізмів є важливим не лише для геймдизайну, але й для економічної освіти та дослідження поведінкової економіки [2].

Проблематика економіки в іграх досліджується у фундаментальній праці В. Лехдонвірти та Е. Кастронови [1], які вводять базові концепції мікроекономіки (ринки, інституції, гроші) в інструментарій геймдизайнера. Окремий напрям становлять дослідження навчальних ігор, занурених у складні економічні системи [2]. Зокрема, К. Дозьє у своїй дисертації аналізує використання інформаційних систем для покращення економічних показників у класичній задачі управління ланцюгами постачання («Beer Game» Дж. Стермана) [2]. Сучасні дослідження також зосереджуються на моделюванні стратегічної невизначеності в іграх з неповною інформацією [3; 4], що безпосередньо стосується прийняття рішень в умовах економічних симуляцій.

Мета роботи. Виявити та систематизувати основні мікро- та макроекономічні механізми, що використовуються у сучасних стратегічних іграх, а також визначити ступінь їх відповідності реальним економічним теоріям.

Результати дослідження. У роботі було проаналізовано чотири популярні стратегічні гри різних піджанрів: StarCraft II (RTS), Civilization VI (TBS), Anno 1800 (симулятор містобудування та економіки) та Victoria 3 (глобальна економічна стратегія) [5]. Ключові результати зведено у табл. 1.

Мікроекономічний рівень в іграх реалізовано через поведінку окремих агентів (робітників, торговців, підприємств). Кожен агент максимізує свою корисність за обмежених ресурсів, що повністю корелює з неокласичною теорією [1]. Як зазначає К. Дозьє, прості економічні системи здатні генерувати високонелінійні динамічні явища, інтуїтивно неочевидні для управління [2]. У цьому контексті стратегічні ігри виступають ефективними навчальними інструментами, що дають змогу гравцям на практиці засвоїти принципи системного мислення.

Таблиця 1 – Поведінкові економічні патерни у стратегічних іграх

Економічний патерн	**Рівень реалізації**	**Реалізація в комп'ютерній грі**	**Відповідність реальному економічному процесу**
Збір та розподіл ресурсів	Мікро	StarCraft II: управління робітниками на мінеральних лініях	Висока (аналог виробничої функції)
Торгівля та ринкові ціни	Мікро / макро	Anno 1800: динамічні ціни на товари залежно від попиту / пропозиції	Середня (спрощена еластичність)
Оподаткування та бюджет	Макро	Civilization VI: податок на прибуток, торговельні тарифи	Висока (аналог фіскальної політики)
Виробничі ланцюжки	Мікро	Anno 1800: деревина → дошки → меблі	Висока (міжгалузевий баланс)
Інфляція та грошова маса	Макро	Victoria 3: збільшення грошової маси без зростання товарів → зростання цін	Помірна (немає кредитно-банківської системи)

Джерело: сформовано авторами

Макроекономічний рівень включає державне регулювання: у Civilization VI можна встановлювати торговельну політику, у Victoria 3 симулюються глобальні ринки, баланс попиту та пропозиції. Сучасні дослідження з теорії ігор розглядають проблему зворотного зв'язку в умовах стратегічної невизначеності [3].

Отже, більшість стратегічних ігор успішно моделюють базові мікроекономічні механізми (гранична корисність, альтернативна вартість, ринкова рівновага). Макроекономічні моделі в іграх є спрощеними, проте здатні ілюструвати ключові ефекти фіскальної політики. Найбільш реалістичні економічні симулятори (Victoria 3, Anno 1800) можуть використовуватися як навчальні тренажери з економіки.

Список використаних джерел

1. Lehdonvirta V., Castronova E. Virtual Economies: Design and Analysis. Cambridge, MA: MIT Press, 2014. 312 p.

2. Dozier K. Theory, Design and Evaluation of a Learning Object Game Immersed in Complex Systems «LOGICS»: Doctoral dissertation. Claremont, CA: The Claremont Graduate University, 2011. 156 p. URL: https://pqdtopen.proquest.com/pubnum/3487695.html (accessed: 08.04.2026).

3. Koessler F., Pahlke M. Feedback Design in Games with Ambiguity-Averse Players. *Journal of Economic Theory.* 2025. Vol. 225. Article 105987. DOI: 10.1016/j.jet.2025.105987.

4. Ui T. Strategic Ambiguity in Global Games. *Games and Economic Behavior.* 2025. Vol. 149. P. 65–81. DOI: 10.1016/j.geb.2024.11.004.

5. Kraus K. M., Kraus N. M. Tsyfrova ekonomika: trendy ta perspektyvy. Digital Economy: Trends and Perspectives. Kyiv: Ahrosvit, 2021. 280 p.

УДК 004.94

Щербань І. О., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Бабаков Р. М., д-р техн. наук, доцент, професор кафедри інформаційних технологій

АЛГОРИТМИ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ЛАНДШАФТІВ ТА СИМУЛЯЦІЇ ЕКОСИСТЕМ

Донецький національний університет імені Василя Стуса, м. Вінниця

Процедурна генерація вмісту (Procedural Content Generation, PCG) – це сукупність алгоритмічних методів створення цифрових даних із мінімальним втручанням людини [1; 4]. У моделюванні середовища PCG є фундаментом для створення динамічних світів, що змінюються в реальному часі [3].

Актуальність роботи зумовлена створенням сучасних тривимірних світів, які вимагають значних часових та людських ресурсів, що призводить до «проблеми гори контенту» (Mountain of Content Problem) [1]. Традиційні методи ручного моделювання не здатні забезпечити необхідний рівень деталізації для масштабних проєктів без експоненціального зростання витрат. Процедурна

генерація дає змогу автоматизувати цей процес, забезпечуючи високу повторюваність ігрового процесу та можливість створення нескінченних ландшафтів. Окрім розважальної індустрії, такі методи є критично важливими для наукових симуляцій та систем віртуальної реальності, де необхідна швидка генерація геологічно достовірних середовищ на основі заданих параметрів [1; 4].

Протягом останніх десятиліть процедурне моделювання перейшло від простих математичних моделей до складних фізико-біологічних симуляцій. Основними методами є стохастичні, симуляційні та агентні підходи, де стохастичні алгоритми на основі фрактального шуму (Perlin, Simplex) досі широко використовуються для швидкого прототипування, хоча їх критикують за відсутність геологічної реалістичності [4]. Сучасні дослідження також акцентують на інтеграції рельєфу з дорогами та будівлями, використанні гідрологічних моделей для створення природної структури ландшафту та підвищенні продуктивності генерації за допомогою GPU для систем реального часу.

Отже, актуальним є проведення комплексного дослідження алгоритмічних основ процедурної генерації ландшафтів та методів симуляції екосистем для створення реалістичних віртуальних середовищ із заданими фізичними та біологічними характеристиками.

У межах дослідження передбачене розв’язання таких завдань:

1) порівняльний аналіз алгоритмів генерації висотних карт на основі випадкових процесів та фрактальних функцій;

2) вивчення механізмів водної та термічної ерозії як засобів уточнення форми рельєфу;

3) дослідження методів процедурного розподілу флори з урахуванням параметрів ґрунту, нахилу поверхні та кліматичних зон;

4) розгляд можливостей автоматизації створення екосистем у сучасних інструментах розробки.

Процес генерації ландшафту розпочинатиметься з формування базової карти висот (heightmap). Для цього буде використано шум Перліна або симплекс-шум, що забезпечує плавні переходи значень та імітацію природних нерівностей [4]. Для підвищення реалістичності рельєфу застосовуватимуться фрактальні методи, зокрема дробовий броунівський рух (fBm), який дає змогу деталізувати поверхню шляхом накладання кількох шарів шуму різної частоти й амплітуди [4]. Такий підхід дасть змогу провести порівняльний аналіз алгоритмів генерації висотних карт.

Наступним етапом стане дослідження фізичної симуляції ерозії для досягнення геологічної достовірності рельєфу. Гідравлічна ерозія моделюватиме рух води, перенесення осадів та їх накопичення в низовинах [5]. Зокрема, буде розглянуто алгоритм Келлі для формування дренажних систем [5]. Також планується використання моделі Женво, заснованої на концепції «конструкційного дерева» [2], що забезпечить автоматичне коригування рельєфу та усунення проблеми «локальних мінімумів».

Симуляція екосистем зосереджуватиметься на автоматизованому розміщенні рослинності відповідно до параметрів середовища. Тип і щільність флори визначатимуться висотою над рівнем моря, нахилом та експозицією схилу, а

також рівнем вологості [1; 2]. Це дасть змогу дослідити методи процедурного розподілу флори залежно від характеристик рельєфу та кліматичних умов.

Для моделювання морфології рослин застосовуватимуться L-системи як засіб формального опису процесів росту [1]. До того ж буде реалізовано алгоритми випадкового розподілу точок із контролем колізій, зокрема диск Пуассона, що забезпечить природну щільність насаджень і запобігатиме накладанню біологічних об'єктів [1]. Автоматизація цих процесів допоможе оцінити можливості сучасних інструментів розробки для створення екосистем.

Процедурна генерація ландшафтів та симуляція екосистем пройшли еволюцію від випадкових математичних функцій до складних систем, що враховують закони фізики та біології. Використання гідрологічних моделей та алгоритмів ерозії дає змогу створювати віртуальні світи, які практично не відрізняються від реальних під час погляду здалеку. Водночас застосування екологічних правил для розподілу рослинності додає ландшафту необхідної деталізації та життєвості. Перспективним напрямом подальших досліджень є поєднання процедурних методів з нейронними мережами (GAN), що дасть змогу генерувати контент на основі стилістики реальних супутникових знімків, зберігаючи водночас повний контроль над топологією світу.

Список використаних джерел

1. Freiknecht J., Effelsberg W. A Survey on the Procedural Generation of Virtual Worlds. *Multimodal Technologies and Interaction.* 2017. Vol. 1, № 4. P. 27.

2. Terrain Generation Using Procedural Models Based on Hydrology / J.-D. Genevaux, E. Galin, E. Guérin, A. Peytavie, B. Benes. *ACM Transactions on Graphics.* 2013. Vol. 32, № 4. P. 143:1–143:13.

3. A Survey of Procedural Methods for Terrain Modelling / R. M. Smelik, K. J. de Kraker, S. A. Groenewegen, T. Tutenel, R. Bidarra. *Proceedings of the CASA 2009 Workshop on 3D Advanced Media In Gaming And Simulation.* 2009.

4. Piispa K. Methods for Procedural Terrain Generation: Master of Science Thesis. University of Turku, 2022. 61 p.

5. Kelley A. D., Malin M. C., Nielson G. M. Terrain Simulation Using a Model of Stream Erosion. ACM SIGGRAPH Computer Graphics. 1988. Vol. 22, № 4. P. 263–268.

СЕКЦІЯ 3
ТЕХНОЛОГІЇ ЗБОРУ, ПРЕДСТАВЛЕННЯ, ОБРОБКИ, ЗБЕРІГАННЯ ІНФОРМАЦІЇ В СУЧАСНИХ ІНФОРМАЦІЙНИХ ТА КОМП'ЮТЕРНИХ СИСТЕМАХ

УДК 004.9:004.421.2

Бондар А. Б., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ МЕТОДІВ ОБЧИСЛЕНЬ У ВІДТВОРЕННІ ДИНАМІКИ ПЕРСОНАЖІВ ВІДЕОІГОР

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасна ігрова індустрія є однією з найдинамічніших сфер цифрових технологій. Еволюція від простих аркад до складних інтерактивних світів вимагає від розробників не лише художнього хисту, а й застосування фундаментальних методів обчислень. Динаміка персонажа – це не просто набір записаних рухів, а складна математична модель, що функціонує в реальному часі. Використання методів обчислень дає змогу створювати персонажів, які не просто існують у просторі, а фізично взаємодіють із ним, реагуючи на гравця та оточення за законами математичної логіки.

Одним із ключових аспектів успішної гри є її дизайн та атмосфера, де кожен компонент впливає на загальний досвід користувача. Важливу роль тут відіграють ігрові механіки, зокрема рух персонажа та його взаємодія з об'єктами. Для того, щоб ці механіки мали природний вигляд, розробники використовують метод інверсної кінематики (ІК). З наукового погляду, це завдання зводиться до розв'язання систем нелінійних рівнянь для визначення координат суглобів персонажа. Оскільки аналітичне розв'язання таких систем потребує значних ресурсів, застосовуються ітераційні чисельні методи (наприклад, метод FABRIK). Це дає змогу персонажу автоматично підлаштовувати положення ніг під нерівний ландшафт або коректно простягати руку до предметів, що значно підвищує рівень реалістичності.

Сучасні ігрові рушії, зокрема Godot, надають потужні інструменти для реалізації такої фізики завдяки сцено-орієнтованій архітектурі та модульному підходу. Це дає змогу представляти динамічні елементи образу, як-от волосся, плащі чи м'які тканини, у вигляді незалежних вузлів, що тестуються окремо. Для відтворення їх руху використовується модель мас-пружин, де об'єкт дискретизується на сітку точок. Динаміка цих точок обчислюється за допомогою методів чисельного інтегрування (наприклад, методу Верле). Це забезпечує стабільність симуляції: плащ персонажа буде реалістично розлітатися під час бігу та плавно зупинятися під дією сили тертя та опору повітря, що розраховуються ітераційно в кожному кадрі.

Окрему увагу приділено системі зіткнень та організації ігрового середовища. Використання вузлів типу TileMap та CollisionShape дає змогу ефективно будувати структуру рівнів, де методи обчислень постійно перевіряють умови перетину геометричних об'ємів. Для оптимізації цього процесу застосовуються методи апроксимації: комп'ютер не обчислює точну фізику для об'єктів, що

знаходяться поза зоною видимості або далеко від гравця, використовуючи натомість спрощені обчислювальні моделі (LOD). Це дає змогу підтримувати стабільну частоту кадрів навіть у складних сценах із багатьма динамічними об’єктами.

Написання скриптів на мові GDScript допомагає гнучко налаштовувати ці складні математичні процеси. Поєднання теоретичних підходів до дизайну гри як системи взаємопов’язаних елементів та практичних інструментів обчислювальної математики створює основу для розробки якісних ігрових продуктів. Отже, методи обчислень виступають невидимим інструментом, що «оживляє» цифрові моделі, перетворюючи їх на динамічні образи, здатні до складної взаємодії у віртуальному просторі.

Список використаних джерел

1. Cario E. Press Start: The History of Video Gaming. Ilex Press, 2025. 288 p.
2. Schell J. The Art of Game Design: A Book of Lenses. CRC Press, 2019. 652 p.
3. Bradfield C. Godot 4 Game Development Projects. Packt Publishing, 2023. 264 p.
4. Документація Godot Engine українською мовою. URL: https://docs.godotengine.org/uk/4.x/index.html
5. Глибовець М. М., Приходько С. О. Методи та алгоритми моделювання фізичних процесів у тривимірній комп’ютерній графіці. *Наукові записки НаУКМА. Комп’ютерні науки.* 2018. Т. 1. С. 12–18.
6. Рижкова Л. В. Застосування ітераційних методів у задачах інверсної кінематики для систем реального часу. *Сучасні інформаційні системи.* 2021. № 5(2). С. 45–51.

УДК 004.2

Бугайов О. С., здобувач вищої освіти 2 курсу спеціальності 122 Комп’ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

МЕТОД ЦЕНТРАЛЬНОЇ РІЗНИЦІ ЧИСЕЛЬНОГО ДИФЕРЕНЦІЮВАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Чисельне диференціювання є важливим розділом методів обчислень, оскільки в багатьох прикладних задачах похідну функції неможливо або недоцільно знаходити аналітично. На практиці функція може бути задана таблицею експериментальних значень, результатами моделювання або складною програмною процедурою. У таких випадках похідну наближено обчислюють за скінченними різницями. Одним із найпоширеніших і найточніших базових підходів є метод центральної різниці [1; 2].

Ідея методу полягає у використанні значень функції по обидва боки від точки, у якій потрібно знайти похідну. Якщо функція $f(x)$ достатньо гладка в околі точки x, то першу похідну можна наближено обчислити за формулою $f'(x)$ приблизно дорівнює $(f(x+h) - f(x-h))/(2h)$, де h – малий крок сітки.

На відміну від правої або лівої різниці, центральна різниця враховує симетричну інформацію про поведінку функції навколо точки x, тому зазвичай забезпечує вищу точність.

Математичне обґрунтування методу випливає з розкладу функції в ряд Тейлора. Для значень $f(x+h)$ і $f(x-h)$ маємо два розклади, у яких парні члени під час віднімання скорочуються. У результаті отримуємо формулу $f(x+h) - f(x-h) = 2hf'(x) + (h3/3)f'''(x) + \ldots$ Після ділення на $2h$ видно, що похибка наближення має порядок $O(h^2)$. Це означає, що під час зменшення кроку h удвічі теоретична похибка методу приблизно зменшується в чотири рази, якщо вплив округлення ще не є суттєвим [2].

Алгоритм застосування методу центральної різниці можна подати так: вибрати точку x, у якій потрібно оцінити похідну; задати крок h; обчислити значення функції $f(x+h)$ та $f(x-h)$; підставити ці значення у формулу центральної різниці; за потреби повторити обчислення для меншого кроку та порівняти результати. Якщо функція задана таблицею, важливо, щоб точки $x-h$ та $x+h$ належали доступній сітці або щоб значення функції в них можна було отримати інтерполяцією.

Розглянемо приклад для функції $f(x) = sin\ x$ у точці $x = 1$. Нехай $h = 0{,}1$. Тоді $f'(1)$ приблизно дорівнює $(sin\ 1{,}1 - sin\ 0{,}9)/0{,}2$. Обчислення дає значення приблизно 0,5394. Точне значення похідної дорівнює $cos\ 1$ приблизно 0,5403. Отже, навіть для порівняно великого кроку центральна різниця дає достатньо близький результат. Якщо зменшити h, похибка спочатку зменшується, але за умови надто малого кроку можуть накопичуватися похибки округлення.

Перевагами методу центральної різниці є простота реалізації, симетричність формули, другий порядок точності та можливість застосування до таблично заданих даних. Метод широко використовують у фізиці, інженерних розрахунках, комп'ютерному моделюванні, аналізі сигналів та чисельному розв'язуванні диференціальних рівнянь. Він також є зручним навчальним прикладом, який показує зв'язок між похідною, скінченними різницями та точністю чисельних алгоритмів [3].

Водночас метод має певні обмеження. Він потребує значень функції з обох боків від точки, тому його незручно застосовувати на межах відрізка без додаткових формул. До того ж вибір кроку h є критичним: занадто великий крок збільшує похибку апроксимації, а занадто малий може посилити вплив машинного округлення. Для зашумлених експериментальних даних центральна різниця може підсилювати коливання, тому іноді її поєднують зі згладжуванням або апроксимацією даних.

Отже, метод центральної різниці є ефективним способом наближеного обчислення похідної. Його головна цінність полягає у поєднанні простої формули та підвищеної точності порівняно з односторонніми різницями. За правильного вибору кроку цей метод дає надійні результати й залишається одним із базових інструментів чисельного диференціювання.

Список використаних джерел

1. Burden R. L., Faires J. D. Numerical Analysis. Boston: Cengage Learning, 2016. 888 p.

2. Chapra S. C., Canale R. P. Numerical Methods for Engineers. New York: McGraw-Hill Education, 2015. 992 p.
3. Sauer T. Numerical Analysis. Boston: Pearson, 2012. 608 p.
4. Quarteroni A., Sacco R., Saleri F. Numerical Mathematics. Berlin: Springer, 2007. 654 p.

УДК 004.422.5

Вовк В. С., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

РОЛЬ ТИПОГРАФІКИ В ДИЗАЙНІ ВЕБРЕСУРСІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сьогодні ми сприймаємо інтернет переважно через текст. Попри велику кількість відеоконтенту та складної графіки, саме шрифти залишаються головним провідником інформації. Типографіка у веб – це не просто вибір «гарних літер», це складний інструмент, який визначає, чи зрозуміє користувач надіслане йому повідомлення, чи відчує він довіру до ресурсу і чи захоче залишитися на сторінці довше, ніж на кілька секунд.

Функціональність та коріння естетики – це коли ми говоримо про функціональність типографіки, варто згадати досвід раннього модернізму. Саме тоді дизайнери почали шукати баланс між естетикою та практичністю. У веб це правило працює бездоганно: шрифт має бути «невидимим» помічником. Якщо користувач помічає шрифт через те, що його важко читати, то дизайн програв. Історичний досвід показує, що типографіка завжди була засобом розв'язання функціональних завдань через візуальну форму [1]. У сучасному веб це трансформувалося у поняття ієрархії, а саме за допомогою розміру, насиченості та відступів ми буквально ведемо око читача від головного до другорядного.

Психологія сприйняття та ідентичність. Людина зчитує настрій сайта ще до того, як почне заглиблюватися в сенс написаного. Шрифти мають свою «харизму». Завдяки правильному підбору гарнітур формується ідентичність бренду, один шрифт може транслювати стабільність і консерватизм, інший – динаміку та інноваційність [2]. Коли типографіка відповідає змісту (наприклад, суворий антикварний шрифт для юридичної фірми або легкий гротеск для стартапу), мозок сприймає інформацію як цілісну та правдиву. Якщо ж виникає дисонанс (веселий дитячий шрифт на сайті банку), підсвідомість сигналізує про небезпеку або некомпетентність ресурсу. На рис. 1 зображено частину доступних сьогодні шрифтів, що ілюструє, наскільки сильно візуальна форма літер може змінювати загальний настрій повідомлення.

Технічна досконалість, адаптивність та рендеринг. Особливість вебдизайну полягає в тому, що ми ніколи не знаємо, з якого пристрою нас будуть читати. Тут на сцену виходить фізика рендерингу та поняття «флюїдної» (плин-

ної) типографіки. Текст має залишатися читабельним і пропорційним як на величезному моніторі, так і на маленькому екрані смартфона [3]. Це вимагає від дизайнера розуміння не лише візуальних правил, а й того, як шрифти взаємодіють із піксельною сіткою екрана. Текст у веб – це жива субстанція, що змінює свою форму, зберігаючи суть.



Рисунок 1 – Візуальне різноманіття типографіки

Типографіка як частина візуальної екосистеми. Зрештою, роль типографіки полягає в інтеграції всіх елементів дизайну в єдину систему. Вона не існує окремо від кольорів чи композиції. Вона є каркасом, на якому тримається весь візуальний досвід користувача [4]. Коли всі аспекти – від інтерліньяжу (відстані між рядками) до кернінгу (відстані між літерами) – налаштовані правильно, інформація сприймається природно, не викликаючи зорової втоми.

Список використаних джерел

1. Родіна Д. В., Костенко І. О. Типографіка як засіб рішення функціональних та естетичних аспектів у дизайні рекламно-поліграфічної продукції раннього модернізму (1910–1935). *III International Science Conference on E-Learning and Education* (Lisbon, Portugal, February 2–5, 2021). Lisbon: International Science Group, 2021. С. 23.

2. Гаврилюк В. Д., Житєньова Н. В. Типографіка в логотипах. Як шрифти формують ідентичність в брендах. *World educational trends: lifelong learning in the information society:* proceedings of the VII International Scientific and Practical Conference (Athens, Greece, October 15–18, 2024). Athens: International Science Group, 2024. С. 33.

3. Лукашук М. М., Зозуля Ю. О. Фізика рендерингу тексту: флюїдна типографіка. VIII International Scientific and Theoretical Conference *«Scientific forum: theory and practice of research»* (San Francisco, USA, March 21, 2025). Primedia eLaunch LLC, 2025. С. 209.

4. Grabska E., Grabska-Gradzińska I., Frodyma T. The Role of Typography in Visual Design. Craft, Technology and Design / eds.: T. Oksala et al. Hämeenlinna: Häme University of Applied Sciences, 2022. P. 118.

УДК 004.422.5

Головня С. О., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

ШТУЧНИЙ ІНТЕЛЕКТ У ВЕБДИЗАЙНІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Однією з найважливіших тенденцій сучасного вебдизайну є активне впровадження штучного інтелекту, який радикально змінює підходи до створення та оптимізації вебсайтів. Цей підхід забезпечує автоматизацію рутинних процесів, прискорення розробки та суттєве підвищення якості користувацького досвіду, даючи змогу дизайнерам зосередитися на творчих і стратегічних аспектах проєктів.

Штучний інтелект (ШІ) у вебдизайні застосовується на різних етапах створення сайтів. ШІ використовує методи машинного навчання, генеративні моделі та аналіз даних для вирішення завдань, які раніше вимагали значних ручних зусиль. Зокрема, ШІ допомагає у генерації візуального контенту, створенні прототипів, аналізі поведінки користувачів та оптимізації інтерфейсів [1; 3]. Ефективність таких рішень часто оцінюється за швидкістю виконання завдань, якістю результату та рівнем автоматизації процесів.

Одним із ключових напрямів є використання генеративного штучного інтелекту для створення зображень, іконок та ілюстрацій. Інструменти типу DALL-E та getimg.ai дають змогу дизайнерам швидко отримувати візуальні елементи за текстовим описом, що значно прискорює етап концептуалізації. До того ж сервіси типу Uizard дають можливість генерувати готові макети та прототипи сайтів на основі простого запиту, скорочуючи час на ручне прототипування [3].

Важливим аспектом є оптимізація користувацького досвіду (UX). Штучний інтелект аналізує поведінку користувачів за допомогою теплових карт, шаблонів кліків та інших метрик, пропонуючи оптимальне розташування елементів сторінки для підвищення конверсії. Інструменти, як-от AI Heatmap у Hostinger Website Builder, допомагають дизайнерам приймати обґрунтовані рішення на основі реальних даних користувачів [3]. Також ШІ активно застосовується для персоналізації контенту, автоматизації написання текстів та забезпечення доступності вебсайтів відповідно до стандартів WCAG [2].

Потужність штучного інтелекту особливо проявляється в роботі вебдизайнерських агентств. За даними досліджень, 93 % дизайнерів уже використовують інструменти ШІ, 58 % – для генерації оригінальних зображень, а 51 % – для створення оптимізованих макетів сторінок. До того ж 82 % розробників застосовують ШІ для допомоги в написанні коду [1].

Водночас використання ШІ супроводжується певними викликами. Серед них – необхідність людського контролю для уникнення упереджень в алгоритмах, питання етики, конфіденційності даних та збереження оригінальності дизайну.

Штучний інтелект не замінює дизайнера, а виступає потужним інструментом, який вимагає грамотної інтеграції з людською креативністю [1–2].

Отже, штучний інтелект є не просто допоміжною технологією, а концептуальною основою сучасного вебдизайну, що поєднує в собі можливості автоматизації рутинних завдань [3], оптимізацію користувацького досвіду [2] та значне прискорення процесів розробки [1], що робить його незамінним інструментом у створенні ефективних, сучасних і користувацько-орієнтованих вебсайтів.

Список використаних джерел

1. Godreau J. How web design agencies use AI (and why you should care). Clutch Resources. URL: https://clutch.co/resources/how-web-design-agencies-use-ai (дата звернення: 24.04.2026).

2. Удовенко О. Революція у вебдизайні: ШІ на передовій – ProIT. ProIT: медіа для профі в IT. URL: https://proit.ua/rievoliutsiia-u-vieb-dizaini-shi-na-pieriedovii/ (дата звернення: 25.04.2026).

3. Козачок А. О. Використання штучного інтелекту у вебдизайні. *Вісник студентського наукового товариства ДонНУ імені Василя Стуса.* 2024. Т. 1, № 16. С. 131–134.

УДК 004.2

Даниленко І. С., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

МЕТОДИ ПРЯМОКУТНИКІВ І ТРАПЕЦІЙ ДЛЯ ЧИСЕЛЬНОГО ОБЧИСЛЕННЯ ІНТЕГРАЛІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Чисельне інтегрування є одним із базових напрямів методів обчислень. Його застосовують тоді, коли первісну функції складно знайти аналітично, інтеграл не має елементарного виразу або функція задана таблицею значень. У таких випадках визначений інтеграл наближено обчислюють за допомогою квадратурних формул. Найпростішими й водночас важливими методами є методи прямокутників і трапецій [1–2].

Геометричний зміст визначеного інтеграла полягає в обчисленні площі під графіком функції на відрізку [a; b]. Основна ідея чисельного інтегрування полягає в тому, щоб замінити цю складну площу сумою простіших фігур. Відрізок інтегрування ділять на n рівних частин із кроком $h = (b - a)/n$, після чого на кожному малому проміжку площу наближують прямокутником або трапецією.

Метод прямокутників використовує значення функції в одній точці кожного підінтервалу. Якщо брати ліві кінці, то формула має вигляд і приблизно дорівнює $h[f(x_0) + f(x_1) + \ldots + f(x_{n-1})]$. Якщо використовувати праві кінці, сума складається зі значень $f(x_1)$, $f(x_2)$, ..., $f(x_n)$. Найчастіше також розгля-

дають метод середніх прямокутників, у якому функцію обчислюють у серединах підінтервалів. Цей варіант зазвичай точніший, оскільки краще враховує поведінку функції всередині кожного проміжку.

Метод трапецій будується на заміні графіка функції на кожному підінтервалі відрізком прямої. Площа під цим відрізком дорівнює площі трапеції, тому загальна формула для рівномірної сітки має вигляд і приблизно дорівнює $h[(f(x_0) + f(x_n))/2 + f(x_1) + f(x_2) + \ldots + f(x_{n-1})]$. На відміну від методу прямокутників, метод трапецій враховує значення функції на обох кінцях підінтервалу, тому для багатьох гладких функцій дає кращий результат [2].

Алгоритм обчислення інтеграла цими методами є простим. Спочатку задають функцію $f(x)$, межі інтегрування a і b та кількість поділів n. Далі обчислюють крок h і вузли сітки. Для методу прямокутників знаходять суму значень функції у вибраних точках і множать її на h. Для методу трапецій окремо враховують крайні значення з коефіцієнтом 1/2, а внутрішні значення додають із коефіцієнтом 1. Збільшення n зазвичай підвищує точність результату.

Розглянемо приклад обчислення інтеграла від $f(x) = x^2$ на відрізку [0; 1]. Точне значення інтеграла дорівнює 1/3 приблизно 0,3333. Якщо взяти $n = 4$, то крок $h = 0{,}25$. Метод лівих прямокутників дає суму 0,21875, метод правих прямокутників – 0,46875, метод середніх прямокутників – 0,328125, а метод трапецій – 0,34375. Під час збільшення кількості поділів усі наближення прямують до точного значення.

Похибка чисельного інтегрування залежить від методу, кроку сітки та властивостей функції. Для лівих і правих прямокутників похибка має перший порядок відносно h, тобто зменшується приблизно пропорційно кроку. Для середніх прямокутників і методу трапецій похибка для гладких функцій має другий порядок. Це означає, що під час зменшення кроку вдвічі похибка в ідеальному випадку зменшується приблизно в чотири рази [3].

Перевагами методів прямокутників і трапецій є простота, наочність і легкість програмної реалізації. Вони добре підходять для навчальних задач, попередніх оцінок, обробки табличних даних і задач, де потрібен швидкий наближений результат. Водночас ці методи можуть бути недостатньо точними для функцій із різкими змінами, розривами або сильними коливаннями. У таких випадках застосовують адаптивні схеми або точніші квадратурні формули.

Отже, методи прямокутників і трапецій є базовими інструментами чисельного інтегрування. Вони демонструють головну ідею наближеного обчислення інтеграла: заміну площі під графіком сумою простих геометричних фігур. Незважаючи на простоту, ці методи залишаються важливими для розуміння чисельних алгоритмів і практичного розв’язання прикладних задач.

Список використаних джерел

1. Burden R. L., Faires J. D. Numerical Analysis. Boston: Cengage Learning, 2016. 888 p.
2. Chapra S. C., Canale R. P. Numerical Methods for Engineers. New York: McGraw-Hill Education, 2015. 992 p.
3. Sauer T. Numerical Analysis. Boston: Pearson, 2012. 608 p.
4. Quarteroni A., Sacco R., Saleri F. Numerical Mathematics. Berlin: Springer, 2007. 654 p.

УДК 004.8:004.9:004.65

Діброва І. С., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Баценко Т. В., канд. техн. наук., доцент, доцент кафедри інформаційних технологій

АНАЛІТИКА ДАНИХ У ВЕБДОДАТКАХ ЗАСОБАМИ POWER BI ТА EXCEL

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні вебдодатки на основі штучного інтелекту активно використовуються для автоматизації створення мультимедійного контенту, зокрема для озвучування тексту та формування відеоматеріалів. Такі рішення є актуальними для освіти, медіа, маркетингу, соціальних мереж та цифрового контент-мейкінгу. Водночас важливим є не лише сам процес створення аудіо- та відеофайлів, а й аналіз даних, які виникають під час роботи системи [1; 5; 7].

У цьому контексті доцільним є використання інструментів аналітики даних, зокрема Microsoft Excel і Power BI. Excel може застосовуватися для первинного збору, очищення, структурування та розрахунку показників, тоді як Power BI забезпечує побудову інтерактивних дашбордів, візуалізацію результатів і моніторинг ключових метрик [2–4]. Поєднання цих інструментів із вебдодатком для автоматичного озвучування тексту та створення відео дає змогу оцінювати продуктивність, стабільність і результативність роботи системи.

Практичною основою дослідження є власний бекенд-проєкт, у якому реалізовано послідовний конвеєр обробки контенту: підготовка тексту, генерація аудіо, створення відеофрагментів і формування підсумкового відеофайла. Для синтезу мовлення використовується API ElevenLabs, а для обробки мультимедійних файлів – FFmpeg [5–6]. Така архітектура дає змогу не лише отримувати готовий відеоконтент, а й накопичувати технічні дані, придатні для подальшого аналізу.

На першому етапі роботи вебдодатка виконується перевірка структури вхідного тексту. Спрощений фрагмент такої перевірки може мати такий вигляд:

```
match = re.match(r'^(\d+)\.', line_strip)

if match:
num = int(match.group(1))
if num != expected_number:
errors.append(f"Помилка в рядку {i}")
```

Цей етап є важливим для валідації даних, оскільки помилки у вхідному тексті можуть впливати на якість подальшої обробки та достовірність аналітичних показників.

Наступним етапом є генерація аудіофайлів на основі текстових фрагментів. Кожен нумерований блок тексту перетворюється на окремий аудіофайл за допомогою API ElevenLabs [5]. Узагальнений фрагмент генерації аудіо можна подати так:

```
stream = client.text_to_speech.convert(
    voice_id=voice_id,
    model_id=model,
 text=text
)

audio_bytes = b"".join(stream)
```

У процесі цього етапу формуються аналітичні параметри, як-от кількість оброблених фрагментів, статус генерації, довжина тексту, тривалість звучання та час виконання операції.

Після створення аудіофайлів система переходить до формування відеофрагментів. Для цього відповідні зображення поєднуються з аудіофайлами за допомогою FFmpeg [6].

На цьому етапі також утворюються важливі дані для аналізу: наявність або відсутність зображення, статус відеогенерації, час обробки, кількість успішних і неуспішних задач. Отже, кожна стадія роботи вебдодатка може бути представлена у вигляді окремого набору показників.

Окреме значення має визначення тривалості аудіофайлів. Цей показник дає змогу обчислювати сумарну тривалість створеного контенту, знаходити найдовші фрагменти та оцінювати співвідношення між довжиною тексту і тривалістю звучання. У проєкті це може бути реалізовано так:

```
audio = File(audio_path)
duration = audio.info.length
```

Надалі такі дані можуть бути використані в Excel для розрахунку агрегованих показників, а в Power BI – для побудови інтерактивних звітів [3; 4].

Для аналітичного оцінювання роботи вебдодатка доцільно використовувати систему ключових показників, наведену в табл. 1.

Наведені дані є умовною аналітичною моделлю, побудованою на основі логіки реального бекенду. Вони дають змогу продемонструвати, як саме можна оцінювати роботу системи засобами аналітики даних. Зокрема, частка успішно згенерованих аудіо становить 94 %, а відео – 89 %, що свідчить про загалом високу результативність системи, але водночас вказує на наявність окремих технічних обмежень на етапі відеогенерації [9].

Аналітичне опрацювання даних дає змогу зробити кілька важливих висновків. По-перше, зі збільшенням довжини тексту зростає тривалість аудіо та час його генерації. По-друге, помилки можуть виникати не лише на етапі синтезу мовлення, а й під час формування відео, наприклад, через відсутність зображення або помилки обробки мультимедійного файла. По-третє, навіть відносно проста система показників дає змогу визначити проблемний етап конвеєра та окреслити напрями вдосконалення системи.

Таблиця 1 – Основні аналітичні показники функціонування вебдодатка

Показник	Значення	Аналітичний зміст
Кількість текстових фрагментів	100	Загальний обсяг вхідних даних
Середня довжина фрагмента, симв.	420	Складність текстової обробки
Успішно згенеровано аудіо, од.	94	Результативність TTS-модуля
Успішно згенеровано відео, од.	89	Результативність відеомодуля
Середній час генерації аудіо, с.	5,6	Продуктивність аудіообробки
Середній час генерації відео, с.	8,4	Продуктивність відеообробки
Середня тривалість, с.	21,3	Обсяг вихідного контенту
Частота помилок, %	6,0	Стабільність функціонування

Отже, вебдодаток для автоматичного озвучування тексту та генерації відео доцільно розглядати не лише як інструмент створення мультимедійного контенту, а й як джерело даних для аналітики. Використання Excel і Power BI дає змогу систематизувати інформацію про роботу сервісу, оцінювати його продуктивність, виявляти слабкі місця та формувати основу для подальшого вдосконалення [2–4]. Це підтверджує практичну цінність поєднання AI-рішень, мультимедійної обробки та BI-інструментів у сучасних вебтехнологіях.

Список використаних джерел

1. Осадчук О. Р. «Розробка системи і інтерфейсу синтезу мовлення українською мовою для сайтів», *Мікросист., Електрон. та Акуст.* Т. 27, вип. 1, 2022. С. 25–29. URL: https://elc.kpi. ua/ article/view/255961

2. Булига К., Толмач М. Візуалізація статистичних даних на платформі Power BI. *Цифрова платформа: інформаційні технології в соціокультурній сфері*. 2021. № 4(1). С. 56–65. https://doi.org/10.31866/2617-796X.4.1.2021.236947

3. Microsoft Learn. Навчальні курси з Power BI: початок роботи з аналітикою даних Microsoft. Офіційна українська документація.

4. Microsoft Learn. Create Dashboards in Power BI. Офіційний навчальний модуль Microsoft Learn, українська локалізація.

5. Microsoft Support. AVERAGE (функція AVERAGE). Офіційна українська довідка Microsoft Excel.

6. Microsoft Support. Використання функції COUNTIF у програмі Microsoft Excel. Офіційна українська довідка Microsoft Excel.

7. Microsoft Support. Функція AVERAGEIF. Офіційна українська довідка Microsoft Excel.

8. Яцолт А. Р., Войтюк М. В. Інформаційна аналітична веб-система прогнозування продаж інтернет-магазину: матеріали конференції ВНТУ. 2022. С. 697–700. URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/40101/15901.pdf?sequence=3&isAllowed=y

9. ElevenLabs. Text to Speech Documentation. Official documentation.

10. Чіома Е. В., Січко Т. В. Машинне навчання в медицині з використанням Power BI EMBEDDED. *Прикладні аспекти сучасних міждисциплінарних досліджень:* матеріали І Всеукр. наук.-практ. конф. (м. Вінниця, 2021). 2021. С. 124–127.

УДК 004.9:004.421.2

Житник Є. Т., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

АНАЛІЗ ВПЛИВУ ПОЧАТКОВОГО НАБЛИЖЕННЯ НА ЗБІЖНІСТЬ ІТЕРАЦІЙНИХ МЕТОДІВ

Донецький національний університет імені Василя Стуса м. Вінниця

Важливе місце в чисельному аналізі та обчислювальній математиці посідають ітераційні методи, оскільки більшість прикладних задач не допускає точного аналітичного розв'язку. Саме тому практична ефективність таких методів прямо залежить від обраного початкового наближення: невдало обране x_0 може збільшити кількість ітерацій обчислення або спричинити розбіжність ітераційного процесу. Метою роботи є дослідження впливу початкового наближення на швидкість та надійність збіжності методу Ньютона на тестових задачах, визначення практичних критеріїв вибору стартової точки та порівняння поведінки методу за різних початкових умов.

Метод Ньютона є одним із найшвидших та найпотужніших ітераційних алгоритмів для знаходження коренів нелінійного рівняння $f(x) = 0$. Ітераційна формула має вигляд:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n = 0,1,2 \dots, \quad (1)$$

де x_0 – початкове наближення;

$f'(x_n) \neq 0$ – значення похідної у поточній точці [1].

Геометричний зміст полягає у знаходженні точки перетину дотичної до графіка функції та осі абсцис у точці наближеного розв'язку. Метод характеризується квадратичною швидкістю збіжності поблизу простого кореня, що робить його надзвичайно ефективним за умови вдалого вибору x_0.

Реалізація методу Ньютона потребує попереднього відокремлення кореня на відрізку [a; b], для якого використовують графічний (побудова графіка), аналітичний (знаходження критичних точок та інтервалів монотонності) та комбінований методи. Для гарантованої та стійкої збіжності ітерації на обраному проміжку мають виконуватися такі умови:

– перша $f'(x)$ та друга $f''(x)$ похідні повинні зберігати сталий знак на проміжку [a; b];

– x_0 обирається на [a; b] за умовою Фур'є: $f(x_0) * f''(x_0) > 0$.

Вона забезпечує наближення до кореня з боку опуклості функції, запобігаючи виходу ітерацій за межі відрізка ізоляції [1]. Чинником, який визначає швидкість та якість ітераційного циклу методу Ньютона, є початкове наближення x_0. У разі правильного вибору стартової точки алгоритм забезпечує квадратичну збіжність (подвоєння кількості значущих цифр на кожному кроці), що дає змогу досягти високої точності за мінімальну кількість ітерацій [2].

Однак, якщо значення x_0 обрано поблизу локального екстремуму, де $f'(x)$ прямує до 0, то вникає ризик розбіжності або зацикленості алгоритму. Дотична стає майже паралельною осі OX, що призводить до порушення монотонності. Обчислення триває до досягнення встановленого критерію зупинки – умови, за якої $|f(x_k)| < \varepsilon$, де $f(x_k)$ – модуль нев'язки, ε – задана точність.

Розглянемо реалізацію методу Ньютона у програмуванні. Проведемо обчислення для різних початкових наближень і обрахуємо кількість ітерацій для кожного x_0. Додатково було реалізовано функцію, яка обчислює x_0 на [a; b] за умовою Фур'є (рис. 1).

```
static double GetInitialGuess(double a, double b, Func<double, double> f, Func<double, double> ddf)
{
    if (f(a) * ddf(a) > 0)
    {
        Console.WriteLine("Результат аналізу: Умова Фур'є виконана для x0 = a");
        return a;
    }

    if (f(b) * ddf(b) > 0)
    {
        Console.WriteLine("Результат аналізу: Умова Фур'є виконана для x0 = b");
        return b;
    }

    Console.WriteLine("Попередження: Умову Фур'є не задоволено на кінцях відрізка.");
    return (a + b) / 2;
}
```

Рисунок 1 – Програмна реалізація алгоритму вибору початкового наближення згідно з умовою Фур'є

Програмна реалізація пошуку кореня рівняння $f(x) = x^3 - x - 2 = 0$ на проміжку [1; 2] наведена на рис. 2.

```
static void TestNewton(double x0, string scenario) {
    Console.WriteLine($">>> {scenario}");
    double xk = x0;
    double eps = 0.0001;
    int k = 0;

    while (k < 20) {
        double fx = F(xk);
        double dfx = DF(xk);

        Console.WriteLine($"Крок {k}: x = {xk:F6}, f(x) = {fx:E2}");

        if (Math.Abs(fx) < eps) {
            Console.WriteLine($"Збіжність досягнута за {k} кроків.");
            return;
        }

        if (Math.Abs(dfx) < 0.001) {
            Console.WriteLine("Критична помилка: f'(x) близька до нуля! Процес розбігається.");
            break;
        }

        xk = xk - fx / dfx;
        k++;
    }
    Console.WriteLine("Висновок: Метод розбігається або потребує занадто багато кроків.");
}
```

Рисунок 2 – Програмна реалізація ітераційного циклу методу Ньютона з контролем критеріїв зупинки

Порівняння результатів обчислювального експерименту для різних сценаріїв вибору x_0 наведено на рис. 3.

```
Визначення точки старту на [1; 2] на основі умови Фур'є
Результат аналізу: Умова Фур'є виконана для x0 = b
>>> Вдале наближення (x0=2)
Крок 0: x = 2.000000, f(x) = 4.00E+000
Крок 1: x = 1.636364, f(x) = 7.45E-001
Крок 2: x = 1.530392, f(x) = 5.39E-002
Крок 3: x = 1.521441, f(x) = 3.67E-004
Крок 4: x = 1.521380, f(x) = 1.74E-008
Збіжність досягнута за 4 кроків.

Задання критичної точки за межами відрізку
>>> Невдале наближення (x0=0.6)
Крок 0: x = 0.600000, f(x) = -2.38E+000
Крок 1: x = 30.400000, f(x) = 2.81E+004
Крок 2: x = 20.274701, f(x) = 8.31E+003
Крок 3: x = 13.529060, f(x) = 2.46E+003
Крок 4: x = 9.039478, f(x) = 7.28E+002
Крок 5: x = 6.059195, f(x) = 2.14E+002
Крок 6: x = 4.094799, f(x) = 6.26E+001
Крок 7: x = 2.825803, f(x) = 1.77E+001
Крок 8: x = 2.053060, f(x) = 4.60E+000
Крок 9: x = 1.657986, f(x) = 9.00E-001
Крок 10: x = 1.533836, f(x) = 7.48E-002
Крок 11: x = 1.521497, f(x) = 6.99E-004
Крок 12: x = 1.521380, f(x) = 6.31E-008
Збіжність досягнута за 12 кроків.
```

Рисунок 3 – Порівняння результатів обчислювального експерименту для різних сценаріїв вибору x_0

Отже, результати підтвердили, що ефективність методу Ньютона залежить від вибору початкового наближення, оскільки саме воно впливає на швидкість та точність результату. Впровадження автоматизованої перевірки умови Фур'є $f(x_0) * f''(x_0) > 0$ гарантує квадратичну швидкість збіжності. Результати реалізації коду на мові C# продемонстрував необхідність коректного вибору початкового наближення. Під час вибору x_0 за умовою Фур'є збіжності було досягнено всього за 4 ітерації, тоді як під час некоректного наближення кількість кроків зросла до 12.

Список використаних джерел

1. Чисельні методи: навч. посіб. / Л. О. Волонтир, О. В. Зелінська, Н. А. Потапова, І. А. Чіков. Вінниця: ВНАУ. 2020. 322 с.

2. Куцмай В. Я. Методи обчислень в розв'язках систем нелінійних рівнянь. Матеріали IV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених *«Прикладні інформаційні технології»:* зб. наук. пр. Вінниця: ДонНУ імені Василя Стуса, 2023. С. 274.

УДК 004.01

Засипка Є. В., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук., старший викладач кафедри інформаційних технологій

ВИКОРИСТАННЯ ЧИСЕЛЬНИХ МЕТОДІВ ОПТИМІЗАЦІЇ У НАВЧАННІ НЕЙРОННИХ МЕРЕЖ

Донецький національний університет імені Василя Стуса, м. Вінниця

Чисельні методи оптимізації у навчанні нейронних мереж – це послідовні математичні дії та правила, за якими поступово змінюються вагові коефіцієнти моделі з метою зменшення помилки передбачення. Ці методи є головним інструментом під час зворотного поширення похибки – саме вони визначають, як швидко мережа навчатиметься і чи зможе вона взагалі знайти прийнятний розв'язок.

Використання чисельних методів оптимізації у навчанні нейронних мереж є основою процесу налаштування ваг. Актуальність цієї теми зумовлена тим, що швидкість збіжності та якість кінцевої моделі безпосередньо залежать від обраного чисельного підходу. Вибір методу оптимізації стає не просто технічною деталлю, а ключовим чинником, що впливає на весь процес розробки глибоких нейронних мереж.

Основним завданням чисельних методів оптимізації є пошук такого набору ваг нейронної мережі, за якого значення функції втрат буде якомога меншим. Вирішення цього завдання дає змогу моделі вловлювати закономірності у вхідних даних і робити правильні прогнози. Процес розв'язання можна подати як послідовність кроків. Першим кроком завжди є обчислення градієнта – вектора, що вказує напрям найшвидшого зростання помилки. Ключовою вимогою тут є правильне застосування обчислення похідних за правилом ланцюжка (так зване зворотне поширення помилки).

Наступна фаза, власне оновлення ваг, є найвідповідальнішою, оскільки саме тут починають діяти різні чисельні схеми. Вона охоплює низку важливих процедур, спрямованих на пришвидшення навчання та уникнення коливань. Необхідно правильно обрати крок навчання: надто малий крок призводить до неприпустимо повільного наближення до мінімуму, а надто великий – взагалі може спричинити розбіжність методу. Застосування методу імпульсу дає змогу згладжувати стрибкоподібні зміни градієнта, накопичуючи зміну ваг у сталому напрямку. Критично важливим є пристосування кроку навчання для кожної ваги окремо – те, що реалізовано в алгоритмах типу Adam чи RMSprop, де попередні значення градієнтів впливають на поточну зміну ваг. Також часто додають так зване затухання ваг (один із видів регуляризації), яке додає до функції втрат штрафний доданок, що стримує надмірне зростання параметрів моделі.

Останнім кроком ітераційного процесу часто є перевірка умови зупинки. Якщо продовжувати навчання після того, як модель уже перестала покращуватись, виникає перенавчання – мережа починає запам'ятовувати випадкові шуми

з навчальних даних замість справжніх закономірностей. Способи зупинки бувають різними: від простої перевірки зменшення помилки на окремій контрольній вибірці до складніших критеріїв, що відстежують зміни ваг або норму градієнта. Варто зазначити, що вибір чисельного методу суттєво залежить від типу нейронної мережі. Для згорткових мереж (тих, що працюють із зображеннями) часто цілком достатньо класичного методу із імпульсом. Для рекурентних мереж (призначених для обробки послідовностей, як-от текст або час) краще підходять пристосовувані методи на зразок Adam, оскільки в них зазвичай виникає проблема згасання градієнта. А для дуже глибоких мереж на основі уваги (трансформерів) останні дослідження показують перевагу спеціальних модифікацій (наприклад, AdamW), які правильніше відокремлюють регуляризацію від основного оновлення ваг.

Окремої уваги заслуговують проблеми, пов’язані із самими чисельними методами. Жоден із методів першого порядку (які використовують лише градієнт) не може гарантувати знаходження найкращого, глобального мінімуму для складних функцій втрат нейронних мереж. У просторих просторах ваг трапляються так звані сідлові точки – у них стандартний метод градієнтного спуску сповільнюється або зупиняється, тоді як пристосовувані методи часто долають такі перешкоди. До того ж обчислювальна складність становить серйозну перешкоду – методи вищого порядку (ті, що намагаються враховувати кривизну поверхні помилок) вимагають надто великих витрат пам’яті та обчислень, тому вони практично не застосовуються для сучасних глибоких мереж із мільйонами параметрів. Отже, навіть найкращий чисельний метод не може забезпечити успіху, якщо будова мережі не відповідає складності задачі або якщо поверхня помилок має незручну форму. Це підкреслює той факт, що вибір методу оптимізації залишається не лише технічним, а значною мірою дослідницьким завданням, де розуміння поведінки різних алгоритмів та досвід часто важать більше, ніж формальні порівняння.

На завершення можна стверджувати, що використання чисельних методів оптимізації є не допоміжним, а головним складником навчання нейронних мереж. Це ітеративний і часто експериментальний вибір схеми оновлення ваг, який перетворює абстрактне завдання мінімізації на практично розв’язну проблему. Подальший розвиток цієї галузі, вочевидь, буде пов’язаний із дедалі більшою автоматизацією вибору як самого методу, так і його налаштувань, а також із розробкою поєднаних (гібридних) підходів, які беруть переваги від різних родин алгоритмів. Вкладання зусиль у вивчення властивостей чисельних оптимізаторів для конкретних типів задач завжди виправдовує себе – це дає змогу будувати швидкі, стійкі та точні нейронні мережі.

Список використаних джерел

1. Гудфеллоу І., Бенджио Й., Курвілль А. Глибоке навчання. Київ: Наукова думка, 2016. 648 с. (переклад англ. вид. 2016).

2. Державний стандарт України. Системи штучного інтелекту. Терміни та визначення. 2014. (На заміну ДСТУ 2860-94). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=25034

3. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach. 2020 P. 702–705.

УДК 004.8:658

Козачок А. О., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Баценко Т. В., канд. техн. наук, доцент, доцент кафедри інформаційних технологій

СУЧАСНІ BI-РІШЕННЯ ДЛЯ ПІДПРИЄМСТВ: ПОРІВНЯЛЬНИЙ АНАЛІЗ МОЖЛИВОСТЕЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

В умовах цифрової трансформації бізнесу та зростання обсягів корпоративних даних ефективне управління інформацією стає стратегічним чинником конкурентоспроможності підприємств. Business Intelligence (BI) – це сукупність технологій, методів і інструментів для збору, обробки та аналізу бізнес-даних з метою підтримки управлінських рішень [1]. Сучасний ринок BI-рішень пропонує широкий спектр платформ – від загальнодоступних хмарних сервісів до спеціалізованих корпоративних систем. Водночас питання вибору оптимальної BI-платформи для конкретного підприємства залишається актуальним і потребує системного порівняльного аналізу.

Питання впровадження BI-систем досліджували вчені B. Weder, S. Negash та ін. [2]. Аналітики компаній Gartner та Forrester Research щорічно публікують звіти про стан ринку BI, зокрема Magic Quadrant для Analytics and Business Intelligence Platforms, що є авторитетним орієнтиром для підприємств. Водночас більшість досліджень не враховує специфіку українського ринку та обмежену бюджетну спроможність малого та середнього бізнесу.

Метою роботи є проведення порівняльного аналізу провідних BI-рішень для підприємств за ключовими функціональними та економічними критеріями. Для досягнення мети поставлено такі завдання:

1. Розглянути теоретичні основи Business Intelligence.
2. Здійснити огляд провідних BI-платформ.
3. Провести їх порівняльний аналіз.
4. Сформулювати практичні рекомендації щодо вибору BI-рішення.

Business Intelligence (BI, бізнес-аналітика) – це комплекс технологій, процесів та інструментів, що забезпечують перетворення «сирих» корпоративних даних на структуровану інформацію для підтримки стратегічних та оперативних рішень [1]. Архітектура сучасного BI-рішення включає такі ключові компоненти: ETL-процеси (Extract, Transform, Load) – для збору та нормалізації даних з різних джерел; сховища даних (Data Warehouse / Data Lake) – для централізованого зберігання; аналітичні двигуни (OLAP) – для багатовимірного аналізу; засоби візуалізації та звітності – для представлення результатів у зручному форматі.

Еволюція BI пройшла кілька етапів: від традиційних інструментів звітності (1990-ті рр.) до self-service BI (2010-ті рр.) та сучасних рішень зі вбудованим штучним інтелектом і предиктивною аналітикою [2].

Microsoft Power BI – одна з найпоширеніших платформ завдяки інтеграції з екосистемою Microsoft 365 та Azure. Пропонує безкоштовну версію (Power BI Desktop) та хмарний сервіс Power BI Service [3].

Tableau (Salesforce) – лідер ринку з потужними можливостями візуалізації та підтримкою великих масивів даних. Орієнтований переважно на великі підприємства та корпоративний сегмент.

Qlik Sense використовує унікальний асоціативний підхід до аналізу даних, що дає змогу виявляти приховані зв'язки між показниками.

SAP BusinessObjects та Oracle Analytics Cloud – корпоративні рішення, інтегровані з відповідними ERP-системами, призначені для великих підприємств.

Looker (Google Cloud) та Metabase – переважно хмарні та open-source-рішення, орієнтовані на технічних користувачів та стартапи.

Для порівняння платформ обрано шість ключових критеріїв: функціональність, інтеграція, масштабованість, безпека, вартість та складність впровадження. Результати наведено в табл. 1.

Таблиця 1 – Порівняльний аналіз BI-платформ

Критерій	**Power BI**	**Tableau**	**Qlik Sense**	**SAP BO**	**Looker**	**Metabase**
Функціональність	Висока	Висока	Висока	Висока	Середня	Середня
Інтеграція	Відмінна (MS)	Широка	Широка	SAP ERP	Google Cloud	Open-source
Масштабованість	Висока	Висока	Висока	Дуже висока	Висока	Середня
Безпека	Висока	Висока	Висока	Дуже висока	Висока	Середня
Вартість	Від $10/міс	Від $70/міс	Від $30/міс	Висока	Від $3 000/р	Безкоштовна
Складність впровадження	Низька	Середня	Середня	Висока	Середня	Низька

Аналіз табл. 1 свідчить, що Microsoft Power BI є оптимальним вибором для малого та середнього бізнесу, який вже використовує інфраструктуру Microsoft. Tableau доцільно обирати для підприємств з розвиненими аналітичними командами, де пріоритетом є якість візуалізації. Для великих корпорацій, інтегрованих із SAP ERP, природним вибором є SAP BusinessObjects. Metabase підходить для стартапів та компаній з обмеженим бюджетом, готових працювати з open-source-рішенням.

Під час вибору BI-платформи підприємству необхідно враховувати: розмір компанії та обсяг даних; наявну IT-інфраструктуру та інтеграційні вимоги; кваліфікацію користувачів (технічних та бізнес-аналітиків); бюджет на впровадження та підтримку; вимоги до безпеки та відповідності нормативним актам [4].

Серед типових помилок під час впровадження BI-систем виділяють відсутність чіткої стратегії та KPI і недостатню підготовку даних перед впровадженням. Проте також можна виділити ігнорування навчання кінцевих користувачів і надмірну кастомізацію. До актуальних тенденцій розвитку ринку BI належать:

вбудований штучний інтелект та NLP-запити; розширення self-service BI; перехід до хмарних та гібридних рішень; зростання ролі real-time-аналітики [5].

Отже, проведений порівняльний аналіз сучасних BI-рішень дає змогу зробити такі висновки. По-перше, вибір BI-платформи не має універсального рішення – він залежить від специфіки підприємства, його IT-інфраструктури та бюджетних можливостей. По-друге, Microsoft Power BI є найбільш збалансованим рішенням за співвідношенням функціональність / ціна для більшості підприємств. По-третє, ринок BI активно розвивається в напрямі інтеграції зі штучним інтелектом та хмарними технологіями. Подальші дослідження доцільно спрямувати на вивчення досвіду впровадження BI-рішень на українських підприємствах.

Список використаних джерел

1. Negash S. Business Intelligence: Підходи до визначення. *Інформаційні системи та технології.* 2019. № 3. С. 45–52.

2. Weder B., Negash S. Business Intelligence and Analytics: From Big Data to Big Impact. *MIS Quarterly.* 2020. Vol. 36, № 4. P. 1165–1188.

3. Microsoft Power BI Documentation. URL: https://learn.microsoft.com/en-us/power-bi/ (дата звернення: 01.05.2026).

4. Мірошниченко О., Боденчук П., Григор'єва Ю. Оптимізація операційної діяльності підприємств через застосування систем BI. *Підприємництво та інновації.* 2024. № 31.

5. Forrester Research. The BI and Analytics Software Market Forecast, 2025 to 2030. Cambridge, 2025. 48 p.

6. Захарова К. В., Січко Т. В. Використання інструменту POWER BI для аналізу фармацевтичного ринку. *Прикладні аспекти сучасних міждисциплінарних досліджень:* матеріали І Всеукр. наук.-практ. конф. (м. Вінниця, 2021). 2021. С. 97–99.

7. Чіома Е. В., Січко Т. В. Машинне навчання в медицині з використанням Power BI EMBEDDED. *Прикладні аспекти сучасних міждисциплінарних досліджень:* матеріали І Всеукр. наук.-практ. конф. (м. Вінниця, 2021). 2021. С. 124–127.

УДК 004.652.4:004.738.5

Кохан Д. Ю., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Ніколюк П. К., д-р фіз.-мат. наук., професор, професор кафедри інформаційних технологій

РЕАЛІЗАЦІЯ ETL-ПАЙПЛАЙНУ ДЛЯ ІНТЕГРАЦІЇ РІЗНОРІДНИХ ДЖЕРЕЛ ДАНИХ У СИСТЕМАХ БІЗНЕС-АНАЛІТИКИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Більшість сучасних підприємств накопичують операційні дані у різних несумісних системах: реляційних базах даних, файлових вивантаженнях і журналах вебаналітики. Узгодження цих джерел для отримання єдиної аналітичної картини потребує значних ручних зусиль і призводить до затримок у прийнятті рішень. Розв'язати цю проблему дає змогу автоматизований ETL-пайплайн, що

об’єднує різнорідні джерела у хмарне сховище з наступною трансформацією та візуалізацією даних [1].

Метою роботи є проєктування та практична реалізація аналітичної платформи для e-commerce на основі стеку Apache Airflow – Google BigQuery – dbt – Power BI, що демонструє повний цикл обробки різнорідних даних.

Джерелами даних слугують чотири компоненти різних форматів: операційна база даних PostgreSQL (25 таблиць, ~65 000 записів, третя нормальна форма), вивантаження сегментів клієнтів з CRM у форматі CSV (2 000 записів), журнал подій вебаналітики у форматі JSON (50 000 подій) та прайс-лист постачальника у форматі XLSX (154 записи).

Таблиця 1 – Характеристика джерел даних аналітичної платформи

Джерело	Формат	Зміст
PostgreSQL	SQL БД	25 таблиць OLTP-моделі: клієнти, товари, замовлення, платежі, відправлення, відгуки (~65 000 записів)
CRM-вивантаження	CSV	Сегменти клієнтів: lifetime value, канал залучення, дата першої покупки (2 000 записів)
Вебаналітика	JSON	Події на сайті: перегляди, додавання до кошика, оформлення покупок (50 000 подій)
Прайс постачальника	XLSX	Собівартість товарів за SKU з датою набуття чинності (154 записи)

Оркестрацію ETL-процесів реалізовано засобами Apache Airflow, розгорнутого у Docker-середовищі. Для кожного джерела розроблено окремий DAG (Directed Acyclic Graph): postgres_to_bq, csv_to_bq, json_to_bq та xlsx_to_bq. DAG для PostgreSQL складається з 25 задач типу PythonOperator з явно визначеними залежностями, що відображають FK-структуру операційної бази даних. На рис. 1 зображено граф виконання DAG postgres_to_bq – завантаження даних з операційної БД PostgreSQL до хмарного сховища; аналогічні за структурою DAGs реалізовано для решти трьох джерел.

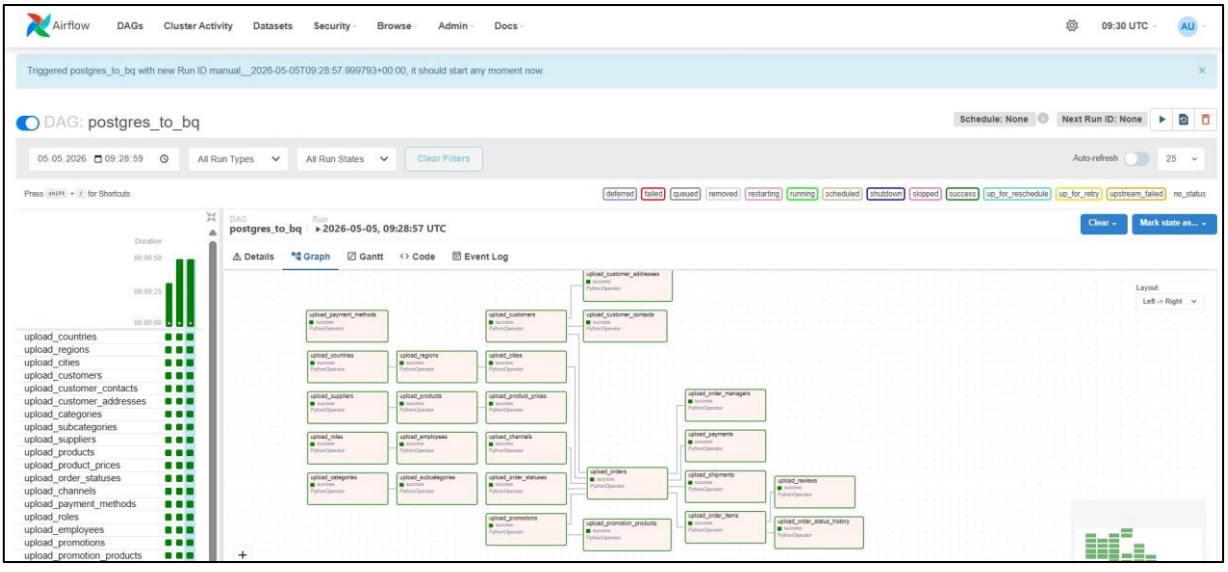


Рисунок 1 – Граф виконання DAG postgres_to_bq в Apache Airflow – завантаження 25 таблиць операційної бази даних PostgreSQL до хмарного сховища BigQuery

Цільовим сховищем даних є Google BigQuery. Сирі дані завантажуються у датасет raw без змін – це забезпечує відтворюваність і можливість повторної обробки. Трансформацію виконує dbt за трирівневою схемою: raw → staging → → marts. На рівні staging 28 SQL-моделей уніфікують типи даних, приводять поля до єдиного іменування та виконують необхідні перетворення (наприклад, усунення конфлікту зарезервованих слів BigQuery). На рівні marts сформовано схему «зірка» з двох таблиць фактів і п’яти таблиць вимірів [2–3].

Таблиця фактів fact_orders (12 519 записів) містить денормалізовані дані про позиції замовлень з розрахованими метриками: загальна сума рядка, розмір знижки, чиста сума, термін доставки. Таблиця фактів fact_web_events (50 000 записів) відображає поведінкові події на сайті – перегляди, додавання до кошика, оформлення та покупки. Таблиці вимірів охоплюють клієнтів (з атрибутами сегментації з CSV-джерела), товари (з показником маржинальності, розрахованим на основі XLSX-прайсу), канали продажу, акції та календар.

Для побудови аналітичного дашборду використано Power BI з підключенням до BigQuery у режимі імпорту. Схему «зірка» у вкладці «Подання моделі» зображено на рис. 2. Зв’язки між таблицями встановлено за ключами-сурогатами, що забезпечує коректну фільтрацію даних у всіх напрямках [4].

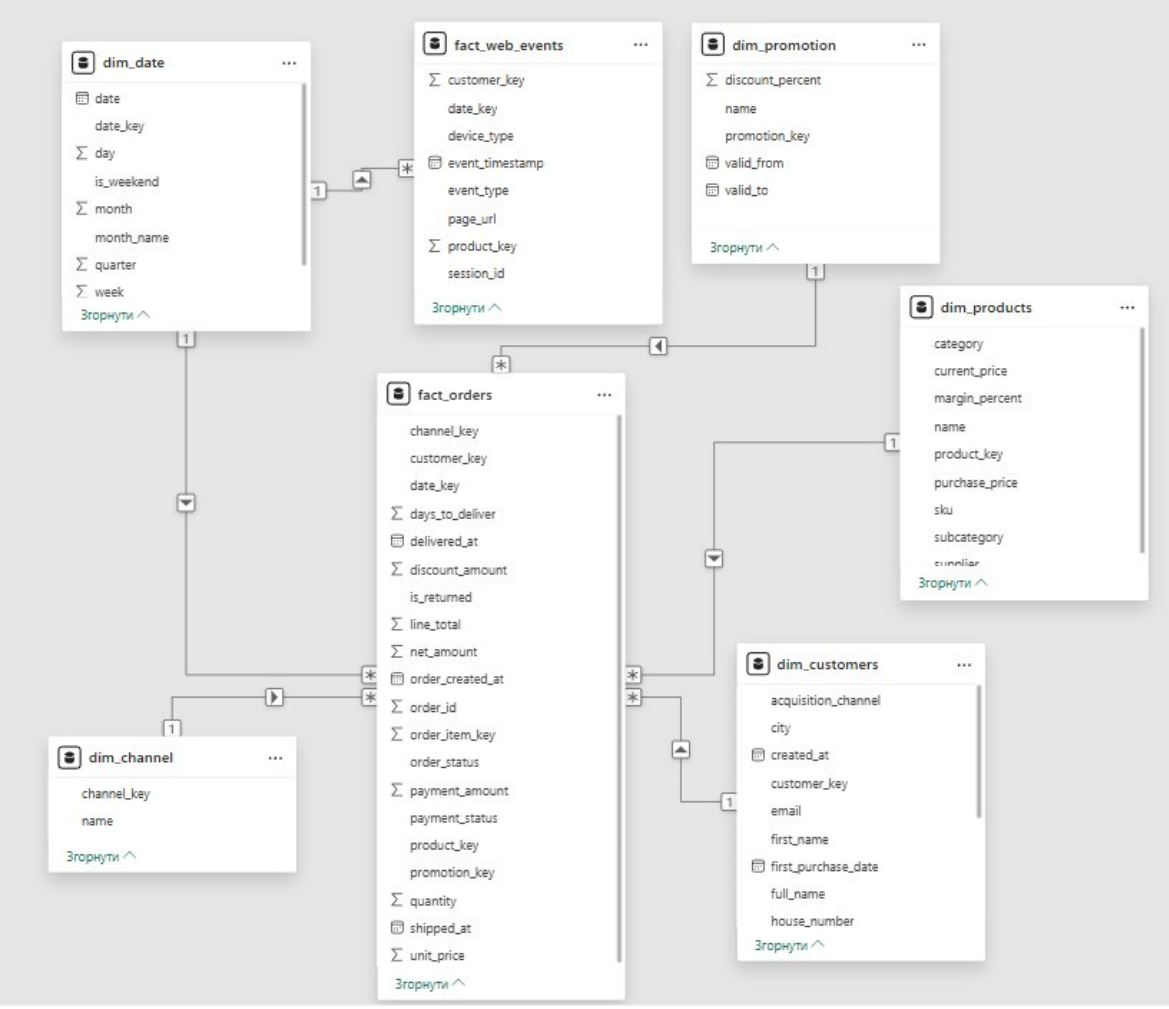


Рисунок 2 – Схема «зірка» аналітичної моделі даних у вкладці «Подання моделі» Power BI – дві таблиці фактів і п’ять таблиць вимірів

Дашборд включає такі аналітичні показники: загальний обсяг продажів, середній чек, відсоток повернень, частка замовлень із акціями, середній термін доставки, конверсія вебподій у покупки за каналами залучення та пристроями, динаміка продажів за місяцями і кварталами, топ-категорії та товари за виручкою і маржинальністю, сегментація клієнтів за показником LTV.

Отримані результати підтверджують, що запропонована архітектура забезпечує повний цикл обробки різнорідних джерел даних: від збору через оркестровані DAG-пайплайни до готових аналітичних метрик в інтерактивному дашборді. Інтеграція CSV-, JSON- та XLSX-джерел разом з операційною базою даних дала змогу збагатити аналітичну модель атрибутами, які відсутні у жодному окремому джерелі: сегментом клієнта, собівартістю товару та поведінковими характеристиками відвідувачів сайта.

Список використаних джерел

1. Munappy A. R., Bosch J., Olsson H. H. Data Pipeline Management in Practice: Challenges and Opportunities. Lecture Notes in Computer Science. 2020. Vol. 12562. P. 168–184. DOI: 10.1007/978-3-030-64148-1_11.

2. Leveraging the Data Lake: Current State and Challenges / C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang. *Lecture Notes in Computer Science.* 2019. Vol. 11708. P. 179–188. DOI: 10.1007/978-3-030-27520-4_13.

3. dbt documentation: How dbt works. dbt Labs. 2024. URL: https://docs.getdbt.com/docs/introduction

4. Putrama I. M., Martinek P. Heterogeneous data integration: Challenges and opportunities. *Data in Brief.* 2024. Vol. 56. Article 110853. DOI: 10.1016/j.dib.2024.110853.

5. Data Lakes: A Survey of Functions and Systems / R. Hai, C. Koutras, C. Quix, M. Jarke. *IEEE Transactions on Knowledge and Data Engineering.* 2023. Vol. 35, № 12. P. 12571–12590. DOI: 10.1109/TKDE.2023.3270101.

УДК 004.42

Маланчук О. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Бабаков Р. М., д-р техн. наук, доцент, професор кафедри інформаційних технологій

АВТОМАТИЗАЦІЯ ПРОЦЕСІВ ПІДБОРУ ПЕРСОНАЛУ ЗА ДОПОМОГОЮ CRM-СИСТЕМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах розвитку цифрових технологій автоматизація бізнес-процесів стає необхідним складником ефективної діяльності підприємств. Особливо актуальною є автоматизація рекрутингових процесів, оскільки рекрутингові агенції працюють із великими обсягами інформації про кандидатів, вакансії та етапи відбору. Виконання таких операцій вручну призводить до значних витрат часу, підвищує ризик помилок та ускладнює процес прийняття рішень.

Сьогодні існує велика кількість систем автоматизації рекрутингу, які забезпечують централізоване управління кандидатами та вакансіями. Проте більшість із них є складними у використанні, потребують значних фінансових витрат або не адаптовані під конкретні потреби локальних рекрутингових агенцій. З метою вирішення зазначених проблем було розроблено десктопний додаток для автоматизації роботи рекрутингової агенції. Основною метою розробки є створення системи, яка забезпечує ефективне управління даними кандидатів, вакансій та супутніх процесів.

Програмний продукт реалізовано з використанням мови програмування Python та бібліотеки PyQt6 для створення графічного інтерфейсу користувача. Для зберігання даних використовується база даних SQLite, а взаємодія з нею здійснюється за допомогою ORM. У межах розробки реалізовано ведення бази кандидатів, управління вакансіями, пошук і фільтрацію кандидатів, контроль етапів підбору та систему задач.

На рис. 1 представлено головне вікно розробленої рекрутингової системи, яке містить модулі керування кандидатами, вакансіями та інструменти пошуку.

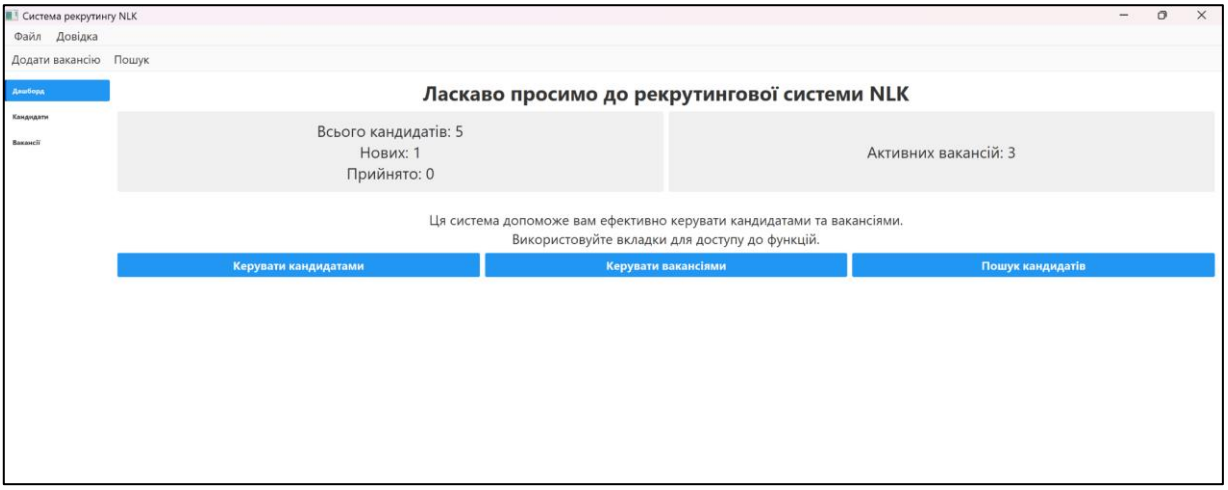


Рисунок 1 – Головне вікно рекрутингової системи NLK

Згідно з архітектурою, наведеною на рис. 2, користувач взаємодіє із системою через графічний інтерфейс, побудований із використанням бібліотеки PyQt6. Усі запити користувача обробляються логікою системи, яка забезпечує виконання бізнес-правил та взаємодію між модулями. Модулі системи відповідають за роботу з кандидатами, вакансіями, задачами та пошуком даних. Збереження інформації здійснюється у локальній базі даних SQLite, що забезпечує швидкий доступ і цілісність даних.

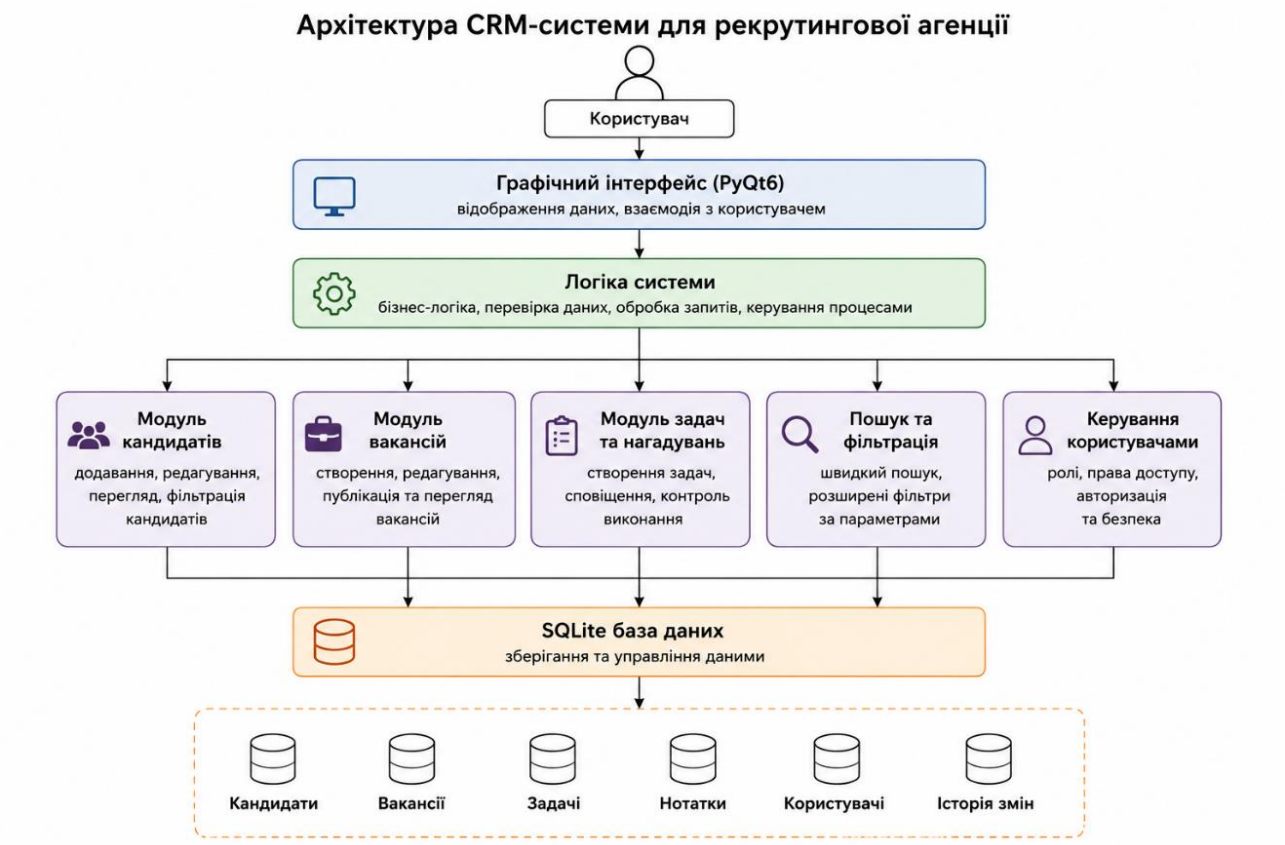


Рисунок 2 – Архітектурна схема CRM-системи для рекрутингової агенції

За результатами досліджень було розроблено десктопний додаток для автоматизації рекрутингових процесів, який відповідає сучасним вимогам до програмного забезпечення. Розроблена система дає змогу підвищити ефективність роботи рекрутингових агенцій, зменшити витрати часу на обробку інформації та покращити якість прийняття рішень.

Список використаних джерел

1. Васильєв О. М. Програмування в Python: теорія і практика. Київ: Ліра-К, 2023.
2. Qt for Python Documentation. URL: https://doc.qt.io/qtforpython-6/
3. SQLite Documentation. URL: https://docs.python.org/3/library/sqlite3.html
4. HURMA HR. ATS-системи для рекрутингу. URL: https://hurma.work/

УДК 004.415.2

Маруняк А. О., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Антонов Ю. С., канд. фіз.-мат. наук, доцент, доцент кафедри інформаційних технологій

ІНТЕГРАЦІЯ ПАТЕРНІВ REPOSITORY ТА UNIT OF WORK У DATA ACCESS LAYER-СИСТЕМИ КЕРУВАННЯ РОЗКЛАДОМ ЗАКЛАДУ ВИЩОЇ ОСВІТИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Системи управління університетським розкладом належать до класу інформаційних систем із підвищеними вимогами до цілісності даних та складною реляційною структурою предметної області [1; 2]. Розклад як доменна сутність не існує ізольовано: кожен запис заняття пов'язаний з аудиторією, викладачем, навчальною групою та дисципліною, а будь-яка операція над ним зачіпає кілька агрегатів одночасно. Саме ця складність предметної області визначила вибір архітектурних рішень на рівні доступу до даних – Data Access Layer, далі DAL – розроблюваної системи.

Метою дослідження є обґрунтування застосування патернів Repository та Unit of Work у DAL інформаційної системи управління розкладом університету на платформі .NET, а також аналіз того, які конкретні архітектурні проблеми вони вирішують у контексті цієї предметної області та в який спосіб реалізуються засобами ASP.NET Core і Entity Framework Core.

Систему побудовано на основі підходу Database-First: схема бази даних була спроєктована заздалегідь, а EF Core використовувався для автоматичної генерації сутностей. Це призвело до появи класів із розгалуженими навігаційними властивостями та надлишковими полями, що відображають структуру реляційних таблиць, але не відповідають потребам бізнес-логіки. Пряма передача таких сутностей між шарами архітектури порушує принцип розмежування відповідальностей і підвищує ризик витоку деталей зберігання на сервісний рівень.

Для вирішення цієї проблеми у DAL реалізовано патерн Repository. Мартін Фаулер визначає його як прошарок між доменними моделями та рівнем відображення даних, що функціонує як колекція об'єктів у пам'яті [3]. На практиці це означає, що автозгенеровані EF Core сутності залишаються виключно у межах DAL, а на сервісний рівень передаються лише Data Transfer Objects, сформовані безпосередньо у репозиторії. Складні SQL-запити, характерні для даної предметної області – пошук вільних аудиторій у заданий проміжок часу з урахуванням типу приміщення, вибірка повного розкладу викладача з урахуванням підгруп, фільтрація занять за корпусом – інкапсульовані всередині відповідних репозиторіїв і не «протікають» у бізнес-логіку. Сервісний рівень отримує готові колекції об'єктів, не маючи жодного уявлення про деталі їх формування.

За аналогією з роботою [4], у системі був реалізованій узагальнений інтерфейс IRepository<T> та базовий абстрактний клас GenericCrudRepository. Сеціалізовані репозиторії – IScheduleRepository, IClassroomRepository, ITeacher WorkloadRepository – розширюють його методами, специфічними для відповідних сутностей, та мають бути імплементовані у конкретних репозиторіях. Офіційна документація Microsoft радить використовувати такий підхід, щоб забезпечити чітке розмежування між бізнес-логікою та деталями зберігання, а також реалізації Unit тестів [5].

Однак ізольованих репозиторіїв виявляється недостатньо, коли одна бізнес-операція зачіпає кілька агрегатів. Характерний приклад у системі розкладу – призначення нового заняття. Ця операція послідовно звертається до кількох репозиторіїв: перевіряє доступність аудиторії через IClassroomRepository, оновлює навантаження викладача через ITeacherWorkloadRepository та зберігає запис заняття через IScheduleRepository. Якщо кожен репозиторій самостійно фіксує зміни у контексті бази даних, будь-який збій на проміжному кроці залишить базу у частково оновленому стані – аудиторія буде позначена як зайнята, але відповідний запис у розкладі так і не з’явиться.

Саме цю проблему вирішує реалізований у системі патерн Unit of Work. Його функція полягає у відстеженні всіх змін, внесених у межах однієї бізнес-транзакції, та гарантуванні їх застосування до бази даних як єдиної атомарної операції або повного скасування у разі виключення [6]. Інтерфейс IUnitOfWork агрегує всі репозиторії та надає єдиний метод фіксації змін – CommitAsync, який викликається один раз після успішного завершення всієї послідовності операцій. Це унеможливлює часткове збереження і гарантує реляційну цілісність незалежно від кількості залучених агрегатів.

Варто зазначити, що Entity Framework Core за своєю природою вже реалізує обидва патерни: DbContext є реалізацією Unit of Work, а DbSet<T> – реалізацією Repository [7]. Проте явне проєктування власних абстракцій IUnitOfWork та IRepository дає принципово важливу перевагу: бізнес-логіка перестає залежати від конкретного ORM-фреймворка. Це означає, що за потреби оптимізувати критичний запит засобами «сирого» SQL або замінити EF Core на інший інструмент – сервісний рівень не потребуватиме жодних змін. До того ж чіткі інтерфейси дають змогу підмінити реальні репозиторії mock-об’єктами під час модульного тестування без розгортання тестової бази даних, що суттєво прискорює CI/CD-процеси.

Інтеграція патернів в ASP.NET Core здійснюється через вбудований механізм Dependency Injection. Інтерфейс IUnitOfWork реєструється з життєвим циклом Scoped, що гарантує існування одного екземпляра DbContext у межах одного HTTP-запиту. Всі репозиторії всередині реалізації UnitOfWork поділяють цей спільний контекст, що і забезпечує можливість атомарної фіксації всіх змін одним викликом CommitAsync.

Отже, вибір патернів Repository та Unit of Work для DAL системи управління розкладом університету є відповіддю на конкретні архітектурні виклики предметної області: необхідність приховати складну логіку багатотабличних вибірок від сервісного рівня, гарантувати транзакційну цілісність під час опе-

рацій над кількома агрегатами та забезпечити незалежність бізнес-логіки від деталей реалізації DAL. Реалізація цих патернів через механізм Dependency Injection у ASP.NET Core перетворює їх із теоретичних шаблонів на практичний фундамент архітектури, що безпосередньо відповідає складності предметної області розроблюваної системи.

Список використаних джерел

1. Антонов Ю. С., Калько Д. Р., Маруняк А. О. Development of a generalized data model and data access architecture for academic scheduling systems using .net and blazor. Scientific practice: modern and classical research methods: *Collection of scientific papers «ΛΟΓΟΣ» with Proceedings of the IX International Scientific and Practical Conference* (Boston, April 10, 2026). Boston-Vinnytsia: Primedia eLaunch & UKRLOGOS Group LLC, 2026. P. 114–117. DOI: 10.36074/logos-10.04.2026.020.

2. Антонов Ю. Реалізація підсистеми керування розкладом університету на основі змішаної архітектури. *Вісник Хмельницького національного університету. Серія: Технічні науки.* 2025. Том 357, № 5.1. С. 13–22. DOI: 10.31891/2307-5732-2025-357-1

3. Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2002. 560 с.

4. Microsoft Learn. Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application. URL: https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions/getting-started-with-ef-5-using-mvc-4/implementing-the-repository-and-unit-of-work-patterns-in-an-asp-net-mvc-application (дата звернення: 01.05.2026).

5. Antonov Y. LST-Web-api URL: https://gitlab.com/antonoveducationprojects/codeexamples/cs/webapi/lst-web-api.git (дата звернення: 04.05.2026).

6. Optimizing Software Architecture: Using the Repository Pattern in Decoupling Data Access Logic. International Scientific Journal of Engineering and Management. 2022. Vol. 01, Issue 01. URL: https://doi.org/10.55041/ISJEM00104 (дата звернення: 03.05.2026).

7. Jakkula V. K. Design Pattern Usage in Large-Scale .NET Applications. International Journal of Engineering and Architecture. 2025. Т. 2, № 2. С. 1–17. URL: https://doi.org/10.58425/ijea.v2i2.420 (дата звернення: 04.05.2026).

УДК 004.942

Носалюк О. І., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ЧИСЕЛЬНИХ МЕТОДІВ ДЛЯ МОДЕЛЮВАННЯ ФІЗИЧНИХ ПРОЦЕСІВ У ВІДЕОІГРАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні відеоігри вимагають високого рівня реалістичності, що неможливо без точного відтворення фізичних законів. Основою для створення таких ігрових механік є фізичні рушії (наприклад, Unity Physics, Havok, Unreal Engine Physics), які використовують чисельні методи для розв'язання диференціальних рівнянь руху в реальному часі.

Основним завданням чисельних методів у геймдеві є обчислення нових положень, швидкостей та прискорень ігрових об’єктів на кожному кадрі. Найпростішим підходом є метод явного Ейлера. Він відрізняється низькою обчислювальною складністю, проте є нестабільним за великих кроків часу, що може призводити до візуальних артефактів (наприклад, провалювання об’єктів крізь текстури або їх неконтрольоване прискорення) [1].

Для підвищення точності та стабільності симуляцій розробники частіше використовують напів’явний метод Ейлера (Symplectic Euler) або метод інтегрування Верле (Verlet integration). Метод Верле особливо ефективний для моделювання систем частинок, поведінки тканин (cloth physics) та регдол-фізики (ragdoll), оскільки він добре зберігає енергію системи та відносно легко обробляє фізичні обмеження (constraints) між з’єднаними об’єктами [2].

У випадках, коли потрібна максимальна точність – наприклад, у реалістичних автосимуляторах або аерокосмічних іграх – можуть застосовуватися методи сімейства Рунге–Кутти (зокрема класичний метод четвертого порядку, RK4). Проте висока математична точність вимагає значних ресурсів процесора, тому такі методи зазвичай ізолюють лише для ключових об’єктів (наприклад, підвіски автомобіля гравця), тоді як фонові об’єкти розраховуються простішими алгоритмами [3].

Головною проблемою застосування чисельних методів у розробці відеоігор є пошук балансу між точністю обчислень та продуктивністю. Гра повинна працювати стабільно (60 або більше кадрів на секунду), тому замість ідеально точних математичних розв’язків використовуються швидкі наближені алгоритми. Для забезпечення детермінованості фізики – щоб об’єкти поводилися однаково на комп’ютерах різної потужності – застосовується фіксований крок часу (fixed time step) для фізичного рушія, не залежний від частоти кадрів графіки [4].

Отже, чисельні методи є фундаментальним математичним інструментом у розробці відеоігор. Правильний вибір алгоритму чисельного інтегрування дає змогу створювати захопливі, динамічні та переконливі ігрові світи, залишаючи водночас достатньо обчислювальних ресурсів для штучного інтелекту та рендеру графіки.

Список використаних джерел

1. Game Physics: Numerical Integration. *Gaffer On Games.* URL: https://gafferongames.com/post/integration_basics/ (дата звернення: 10.05.2026).

2. Verlet Integration in Game Physics. Medium. URL: https://medium.com/@sarvesh.h/verlet-integration-in-game-physics-4a7ef7557a07 (дата звернення: 10.05.2026).

3. Physics in Games: Which Integration Method is Best? *TutsPlus.* URL: https://gamedevelopment.tutsplus.com/tutorials/how-to-create-a-custom-2d-physics-engine-the-core-engine--gamedev-7493 (дата звернення: 10.05.2026).

4. Fix Your Timestep! *Gaffer On Games.* URL: https://gafferongames.com/post/fix_your_timestep/ (дата звернення: 10.05.2026).

УДК 004.42:519.6

Олексієнко Є. О., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки, Павлюк О. А., канд. техн. наук., старший викладач кафедри інформаційних технологій

ЗАСТОСУВАННЯ МЕТОДУ ТРАПЕЦІЙ У МОБІЛЬНИХ ДОДАТКАХ ДЛЯ ТРЕНІНГУ ФІЗИЧНОЇ АКТИВНОСТІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Чисельне інтегрування – це галузь прикладної математики, що зосереджена на знаходженні площі під кривою функції для перетворення дискретних величин у неперервні. Ця технологія широко застосовується у розробці фітнес-додатків для імітації точного відстеження рухів людини. У сучасному цифровому середовищі мобільні додатки на основі чисельних методів відіграють ключову роль, забезпечуючи автоматизацію трекінг-процесів, надання інформації про тренування та персоналізовану взаємодію у режимі реального часу [1; 2].

Основою таких обчислень є метод трапецій. Цей алгоритм дає змогу системі «зчитувати» та накопичувати дискретні показники з акселерометра. Ключовим компонентом є перетворення цих даних на швидкість та відстань, що фокусується на визначенні зміни положення пристрою у просторі та вилученні корисного сигналу. Наприклад, маючи значення прискорення, алгоритм ідентифікує поточну швидкість, а застосувавши метод повторно – пройдену відстань за певний час [3].

Обробка потоку даних (Data Processing) є наступним критичним етапом. Цей компонент відповідає за підтримання точності обчислень та прийняття рішень про те, як саме фільтрувати шум від сенсорів перед інтегруванням. У простих додатках це може бути реалізовано через фіксований крок часу. Однак у складних системах використовуються статистичні моделі (наприклад, фільтр Калмана) або згладжувальні алгоритми для вибору оптимального набору даних [4].

Важливим аспектом є розрахунок витраченої енергії, тобто процес, за допомогою якого додаток формулює загальну статистику тренування. Ранні версії додатків покладалися на заздалегідь прописані шаблони, куди просто підставлялися базові значення з крокоміра. Сучасні алгоритми здатні динамічно адаптувати крок інтегрування, створюючи більш точні та гнучкі графіки активності [4].

Справжній прорив у можливостях фітнес-додатків пов'язаний з появою комбінованих моделей та алгоритмів розпізнавання активності (HAR). Ці моделі, навчені на величезних масивах даних з акселерометрів, здатні розуміти контекст руху, вловлювати специфіку бігу чи їзди на велосипеді та генерувати надзвичайно точні результати розрахунку дистанції [5].

Сучасні алгоритми також дають змогу додаткам адаптуватися в процесі взаємодії. Використовуючи методи машинного навчання, система може аналі-

зувати патерни рухів користувачів для коригування обчислень площі під кривою. Це дає змогу персоналізувати тренування та з часом покращувати якість обробки даних чисельного інтегрування [4].

Однією з основних проблем використання методу трапецій у додатках є висока частота опитування сенсорів та пов'язане з цим енергоспоживання за зменшення кроку обчислень. Вони вимагають значних апаратних ресурсів смартфона. Іншою серйозною проблемою є схильність чисельного інтегрування до «дрейфу» – накопичення помилки та генерування правдоподібної, але фактично неправильної інформації про пройдений шлях. Для боротьби з цим застосовуються новітні архітектури адаптивного кроку, які «заземлюють» обчислення на показах педометрів та модулів GPS [3].

Отже, алгоритми чисельного інтегрування є ядром сучасних додатків для тренінгу, що перетворює їх на потужні інструменти для моніторингу активності. Їх правильне застосування забезпечує реалізацію інтелектуальних, точних та корисних систем відстеження здоров'я, що є ключовим аспектом сучасної цифрової епохи.

Список використаних джерел

1. Validity and Reliability of a Smartphone Accelerometer for Measuring Lift Velocity in Bench-Press Exercises. URL: https://surl.li/lojugs (дата звернення: 04.05.2026).

2. A Comprehensive Analysis on Wearable Acceleration Sensors in Human Activity Recognition. URL: https://surl.li/uyivtd (дата звернення: 04.05.2026).

3. Inertial Sensor Algorithm to Estimate Walk Distance. URL: https://surl.li/edlpnq (дата звернення: 04.05.2026).

4. Gait metrics analysis utilizing single-point inertial measurement units: a systematic review. URL: https://surl.li/utxxgs (дата звернення: 21.10.2025).

5. Wearable Monitoring Devices for Biomechanical Risk Assessment at Work: Current Status and Future Challenges. URL: https://surl.li/ihtryc (дата звернення: 21.10.2025).

УДК 004.942:519.23

Черніхевич О. В., здобувачка вищої освіти 3 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ БУТСТРЕПУ ТА ПЕРЕХРЕСНОЇ ПЕРЕВІРКИ ДЛЯ ОЦІНКИ МОДЕЛЕЙ НАБОРУ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Створення надійних алгоритмів машинного навчання неможливе без адекватної перевірки їх здатності до узагальнення. Якщо ми зосередимося виключно на мінімізації емпіричного ризику на тренувальних даних, то майже гарантовано зіткнемося з перенавчанням (overfitting). Замість того, щоб знаходити реальні приховані закономірності, модель почне запам'ятовувати статистичний шум.

Щоб уникнути цього та об’єктивно оцінити справжній (генералізований) ризик, використовують методи повторної вибірки (ресемплінгу). Вони чудово імітують поведінку моделі під час зустрічі з абсолютно новими даними [3; 4]. Сьогодні фахівці часто працюють із неідеальними датасетами: незбалансованими, занадто малими або нерепрезентативними. Неправильно підібраний метод валідації в таких умовах може серйозно спотворити метрики точності. Саме тому глибоке розуміння та порівняння математичної бази бутстрепу і перехресної перевірки є критично важливим для предиктивної аналітики [5].

Статистичне підґрунтя бутстрепу заклав Б. Ефрон, спочатку довівши його ефективність для розрахунку довірчих інтервалів [1]. У контексті машинного навчання цю тему (поряд із перехресною перевіркою) фундаментально дослідив Р. Кохаві. Саме він на базі емпіричних експериментів довів перевагу 10-кратної перехресної перевірки [2]. Сучасні дослідження переважно зосереджені на адаптації цих методів – наприклад, на стратифікованому ресемплінгу для класифікації у випадках сильного дисбалансу класів [3; 5].

Оцінка якості будь-якої моделі (чи то регресія, чи класифікація) зводиться до пошуку математичного сподівання функції втрат між прогнозом та фактом. Метод k-кратної перехресної перевірки (k-fold CV) вирішує це завдання так: він випадково розбиває генеральний набір даних розміром N на k однакових блоків, що не перетинаються між собою [4]. Далі алгоритм виконує k ітерацій: щоразу один блок (приблизно N/k записів) стає екзаменаційним, а на всіх інших модель вчиться. Загальна оцінка помилки вираховується як середнє значення:

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} L\left(y_i, \hat{f}^{-k}(x_i)\right), \quad (1)$$

де L – обрана метрика помилки;

y_i – справжній відгук;

$\hat{f}^{-k}(x_i)$ – прогноз моделі, що навчалася без i-го фолду [5].

Цікавим крайнім випадком є метод LOOCV (Leave-One-Out Cross-Validation), де $k = N$. Його використання часто супроводжується величезними обчислювальними витратами та високою дисперсією результатів, якщо дані зашумлені. На практиці для класифікації найчастіше використовують стратифіковану версію (Stratified CV). Вона гарантує, що початкові пропорції класів зберігаються в кожному фолді [3].

Як альтернатива виступає метод бутстрепу, заснований на логіці Монте-Карло. Тут нові вибірки формуються шляхом випадкового вибору елементів із поверненням [1]. Розмір кожної такої вибірки дорівнює початковому N. Оскільки вибраний об’єкт повертається назад, ймовірність того, що він узагалі не потрапить до тренувального набору, визначається через асимптотичну границю:

$$\lim_{N\to\infty}\left(1-\frac{1}{n}\right)^{N} = e^{-1} \approx 0{,}368\,. \quad (2)$$

Це означає, що типова бутстреп-вибірка містить лише приблизно 63,2 % унікальних екземплярів. Решта 36,8 % записів формують так звану позапакетну вибірку (Out-of-Bag, OOB), на якій і проводиться тестування. Оскільки модель бачить не всі унікальні дані під час навчання, класичний бутстреп схильний песимістично переоцінювати помилку. Щоб компенсувати це зміщення, застосовують оцінку «.632 бутстреп», яка є лінійною комбінацією помилки на тренувальних даних $\overline{err}$ та OOB-помилку $Err^{(1)}$:

$$Err^{(.632)} = 0{,}368 \cdot \overline{err} + 0{,}632 \cdot Err^{(1)}. \quad (3)$$

Для зручності ми систематизували ключові відмінності методів у табл. 1.

Таблиця 1 – Зіставлення властивостей k-кратної перехресної перевірки та бутстрепу

Критерій оцінювання	***k*-кратна перехресна перевірка**	**Метод бутстрепу (OOB / .632)**
Механізм формування	Розбиття простору ознак без повернення	Випадкова вибірка елементів із поверненням
Статистичне зміщення (Bias)	Дуже низьке (особливо за $k = 10$ та оптимізації)	Відносно високе (залежить від наявності дублікатів)
Дисперсія результатів (Variance)	Висока (якщо кількість навчальних даних мала)	Дуже низька (за рахунок агрегації багатьох вибірок)
Обчислювальне навантаження	Лінійне, пропорційне параметру k	Значне, вимагає генерації B (часто $B > 100$) вибірок
Доцільність використання	Датасети середнього та великого розміру ($N > 200$)	Вкрай малі масиви даних ($N < 100$)

Аналізуючи дані табл. 1, можна сказати, що перехресна перевірка є еталонним інструментом оцінювання завдяки відсутності систематичної похибки [4]. Її головна слабкість – це зростання дисперсії на дуже малих вибірках, адже зміна одного фолду може радикально переінакшити навчальний набір. Бутстреп же успішно згладжує подібні коливання завдяки сотням ітерацій ресемплінгу. Це дає змогу отримати стабільну оцінку навіть за умов гострого дефіциту інформації [2].

Отже, проведений аналіз доводить, що універсального підходу серед методів оцінки не існує:

1. k-кратна перехресна перевірка (зокрема її стратифікована версія за $k = 10$) є справжнім золотим стандартом машинного навчання для вибірок середнього та великого розміру, оскільки забезпечує чесну, незміщену оцінку ефективності.

2. Метод бутстрепу розкриває свій потенціал під час роботи з ультрамалими датасетами. Його здатність кардинально мінімізувати дисперсію повністю перекриває недоліки, як-от висока обчислювальна складність та наявність певного статистичного зміщення.

Список використаних джерел

1. Efron B., Tibshirani R. J. An Introduction to the Bootstrap. Boca Raton: Chapman & Hall/CRC, 1994. 436 p.

2. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence.* Montreal, 1995. Vol. 2. P. 1137–1143.

3. Олещенко Л. М. Машинне навчання: комп'ютерний практикум: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2022. 92 с.

4. Інтелектуальний аналіз даних та машинне навчання. Частина 1. Базові методи та засоби аналізу даних / Я. В. Іванчук, В. І. Месюра, А. А. Яровий, О. Д. Манжілевський. Вінниця: ВНТУ, 2021. 69 с.

5. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York: Springer, 2009. 745 p.

УДК 004.6

Паламарчук Т. Л., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

МЕТОД ДИХОТОМІЇ ЯК СПОСІБ ЧИСЕЛЬНОГО РОЗВ'ЯЗАННЯ РІВНЯНЬ

Донецький національний університет імені Василя Стуса, м. Вінниця

Метод дихотомії є одним із базових чисельних методів, який застосовується для знаходження наближених коренів рівняння $f(x) = 0$. Його використовують тоді, коли точний аналітичний розв'язок отримати складно або недоцільно, а також у випадках, коли потрібно забезпечити надійний контроль похибки. Метод є важливим для курсу «Методи обчислень», оскільки демонструє основний принцип чисельного аналізу: заміну точного розв'язку послідовністю наближень із заданою точністю [1; 2].

Математична основа методу дихотомії пов'язана з теоремою Больцано–Коші. Якщо функція $f(x)$ неперервна на відрізку [a; b], а її значення на кінцях відрізка мають різні знаки, тобто $f(a) \cdot f(b) < 0$, то всередині цього відрізка існує принаймні один корінь рівняння $f(x) = 0$. Саме ця умова є головною передумовою застосування методу. Вона не завжди гарантує єдиний корінь, проте забезпечує існування кореня на вибраному проміжку [2].

Суть методу полягає в послідовному поділі початкового відрізка навпіл. На кожній ітерації обчислюється середина відрізка за формулою $c = (a + b)/2$. Далі визначається значення функції $f(c)$. Якщо $f(a) \cdot f(c) < 0$, то корінь міститься на відрізку [a; c], і праву межу замінюють на c. Якщо ця умова не виконується, то корінь міститься на відрізку [c; b], і ліву межу замінюють на c. Отже, після кожного кроку довжина відрізка, у якому міститься корінь, зменшується удвічі.

Алгоритм методу дихотомії можна подати у вигляді такої послідовності дій: вибрати початковий відрізок [a; b], для якого виконується умова зміни знака; обчислити середину c; визначити знак $f(c)$; вибрати ту половину відрізка, на

кінцях якої значення функції мають різні знаки; повторювати процес до досягнення заданої точності. Найчастіше критерієм завершення обчислень є умова $(b-a)/2 \leq \varepsilon$, де ε – допустима похибка наближення.

Важливою перевагою методу дихотомії є можливість заздалегідь оцінити кількість ітерацій. Якщо початкова довжина відрізка дорівнює $b_0 - a_0$, то після n ітерацій вона становитиме $(b_0 - a_0)/2^n$. Отже, похибка наближення не перевищує половини довжини поточного відрізка. Завдяки цьому метод є передбачуваним і зручним для реалізації в комп'ютерних програмах [3].

Розглянемо приклад застосування методу дихотомії для рівняння $x^3 - x^2 = 0$. На відрізку $[1;\ 2]$ маємо $f(1) = -2,\ f(2) = 4$. Оскільки $f(1) \cdot f(2) < 0$, то на цьому відрізку існує корінь. Перша середина дорівнює $c = 1{,}5$, а $f(1{,}5) = -0{,}125$. Отже, корінь міститься на відрізку [1,5; 2]. Після кількох наступних поділів отримуємо наближене значення кореня $x \approx 1{,}52$. Цей приклад показує, що метод поступово звужує область пошуку та не втрачає корінь за умови правильного вибору початкового відрізка.

До основних переваг методу дихотомії належать простота, надійність, гарантована збіжність за виконання умов і відсутність потреби в обчисленні похідної. Метод зручно використовувати для початкової локалізації кореня або для перевірки результатів, отриманих іншими чисельними методами. Водночас він має й обмеження: швидкість збіжності є відносно невисокою, а для застосування методу обов'язково потрібно знайти відрізок, на кінцях якого функція змінює знак [4].

Отже, метод дихотомії є надійним і математично обґрунтованим способом чисельного розв'язання нелінійних рівнянь. Його головна цінність полягає у простоті реалізації та можливості точного контролю похибки. Незважаючи на те, що метод поступається швидкістю деяким іншим алгоритмам, він залишається важливим інструментом у чисельному аналізі, навчальних задачах і прикладних обчисленнях.

Список використаних джерел

1. Burden R. L., Faires J. D. Numerical Analysis. Boston: Cengage Learning, 2016. 888 p.
2. Chapra S. C., Canale R. P. Numerical Methods for Engineers. New York: McGraw-Hill Education, 2015. 992 p.
3. Sauer T. Numerical Analysis. Boston: Pearson, 2012. 608 p.
4. Atkinson K. E. An Introduction to Numerical Analysis. New York: John Wiley & Sons, 1989. 693 p.

УДК 004.01

Пацула А. О., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук., старший викладач кафедри інформаційних технологій

ОБЧИСЛЮВАЛЬНІ МЕТОДИ РОЗРАХУНКУ НОРМАЛЕЙ ВЕРШИН ДЛЯ АЛГОРИТМІВ ЗГЛАДЖУВАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасній комп'ютерній графіці та геометричному моделюванні візуальна якість тривимірних об'єктів критично залежить від коректного математичного опису їх поверхонь. Згладжування полігональних сіток (mesh smoothing) є фундаментальним етапом у підготовці 3D-моделей для рендерингу, фізичних симуляцій та інтеграції в ігрові рушії. Актуальність теми зумовлена необхідністю забезпечення високої точності обчислень та відсутності візуальних артефактів під час шейдингу складних геометричних форм, що вимагає використання оптимізованих чисельних методів.

Аналіз останніх досліджень показує, що класичні підходи, як-от базове тоноване затінення Гуро або Фонга, заклали основу для інтерполяції кольору та освітленості. Проте для сучасного hard-surface-моделювання стандартного арифметичного усереднення нормалей вже недостатньо. Дослідження в галузі обчислювальної геометрії (зокрема праці Н. Макса) доводять, що ігнорування топологічних особливостей сітки призводить до викривлення освітлення на стиках полігонів різних розмір.

Базовим етапом роботи будь-якого алгоритму згладжування є визначення нормалі до кожної площини (грані) сітки. У тривимірному просторі для трикутника з вершинами A, B та C вектор нормалі грані обчислюється через векторний добуток двох його направляючих ребер: $N = (B - A) * (C - A)$. Однак для забезпечення ефекту неперервної, гладкої поверхні (Soft edges) необхідно розрахувати нормалі для самих вершин.

Найпростішим чисельним методом є незважене усереднення (Unweighted Average), за якого нормаль вершини обчислюється як просте середнє арифметичне нормалей усіх граней, що сходяться у цій вершині. Головним недоліком цього методу є висока чутливість до теселяції: дрібні «сміттєві» полігони впливають на фінальний вектор так само сильно, як і великі площини, що генерує значні артефакти освітлення.

Для вирішення цієї проблеми застосовуються методи зваженого усереднення (Weighted Average). Серед них виділяють два основні:

1. Зважування за площею (Area Weighted): ваговий коефіцієнт кожної грані пропорційний її площі. Це частково усуває вплив дрібних трикутників на загальну геометрію вершини.

2. Зважування за кутом (Angle Weighted): вага пропорційна величині кута між ребрами грані, що сходяться у розрахунковій вершині. З математичного погляду, цей метод є найбільш стабільним і точним для розрахунку гладких

поверхонь, оскільки він базується безпосередньо на кривизні геометрії в конкретній точці простору.

До того ж у сучасних алгоритмах згладжування активно застосовується метод порогового кута (Smooth Angle Threshold). Програмний інструментарій обчислює двогранний кут (dihedral angle) між суміжними полігонами. Якщо цей кут перевищує задане користувачем критичне значення, алгоритм «розщеплює» вершину на кілька незалежних копій (split normals). Кожна копія отримує свій власний вектор нормалі, що призводить до розриву інтерполяції світла і створює математично чітке гостре ребро (Hard Edge).

Отже, вибір обчислювального методу розрахунку нормалей є ключовим фактором у 3D-моделюванні. Найкращі результати демонструє комбінація алгоритмів зважування за кутом (Angle Weighted) та порогового розщеплення нормалей. Попри широкі можливості стандартних PBR-матеріалів, специфічні візуальні задачі часто вимагають написання користувацьких шейдерів мовою GLSL (OpenGL Shading Language). Наприклад, для технічної візуалізації внутрішньої структури об’єкта (ефекти рентгенівського сканування, силових полів або голограми) розробляються кастомні вершинні (Vertex) та фрагментні (Fragment) шейдери. Вони дають змогу маніпулювати позицією та кольором кожного пікселя на основі математичних функцій (наприклад, шуму Перліна або тригонометричних хвиль) та створювати ефекти плавного переходу від суцільної моделі до її полігональної сітки (Wireframe Transition). Важливою перевагою такого підходу є те, що всі математичні обчислення шейдерів виконуються паралельно на потужностях відеокарти (GPU), не блокуючи основний потік виконання JavaScript у браузері.

Важливим викликом під час розробки вебдодатків на базі WebGL є питання доступності (Accessibility, a11y) та пошукової оптимізації (SEO). Оскільки HTML-елемент <canvas>, всередині якого відбувається рендеринг графіки, є абсолютно «сліпим» і непрозорим для пошукових роботів та скринрідерів (Screen Readers), виникає проблема ізоляції контенту. Для вирішення цієї задачі необхідно створювати паралельне семантичне DOM-дерево (Document Object Model). Цей архітектурний патерн передбачає накладання невидимих HTML-елементів із відповідними атрибутами ARIA поверх 3D-сцени, координати яких динамічно синхронізуються з положенням тривимірних об’єктів (наприклад, кнопок на 3D-моделі термінала). Такий підхід гарантує, що вебдодаток коректно індексуватиметься пошуковими системами та залишатиметься доступним для користувачів із порушеннями зору, відповідаючи міжнародним стандартам WCAG (Web Content Accessibility Guidelines).

Логічним продовженням обчислення нормалей на етапі моделювання є їх інтеграція у конвеєр рендерингу WebGL через побудову дотичного простору (Tangent Space). Для того, щоб кастомні фрагментні шейдери могли коректно обробляти освітлення складних hard-surface об’єктів з використанням запечених карт нормалей (Normal Maps), недостатньо мати лише базовий вектор нормалі вершини. Обчислювальний алгоритм вимагає формування ортогональної TBN-матриці (Tangent, Bitangent, Normal). Математично вектор тангенса розраховується на основі UV-координат полігональної сітки, що дає змогу шейдеру трансформувати напрямок світла зі світового простору безпосередньо у локаль-

ний простір поверхні полігону. Це гарантує, що мікрорельєф та фаски будуть коректно відбивати світло під будь-яким кутом обертання моделі у браузері.

До того ж під час розробки подібних рішень критично важливим є профільованість продуктивності (Performance Profiling). Оскільки розрахунки освітлення та нормалей виконуються для кожного пікселя екрана (per-pixel lighting), висока роздільна здатність сучасних моніторів може призвести до падіння частоти кадрів (FPS). Тому в архітектуру вебдодатків закладаються методи обчислювальної оптимізації, зокрема використання алгоритмів відсікання невидимих граней (Backface Culling) та зниження точності розрахунків у шейдерах (від highp до mediump для чисел з плаваючою комою) у тих частинах сцени, які не потребують ідеальної точності нормалей.

Підсумовуючи, можна стверджувати, що забезпечення високої візуальної якості тривимірних об’єктів у сучасних інтерактивних додатках вимагає глибокого математичного та архітектурного підходу. Він починається з коректного обчислення нормалей вершин (за допомогою алгоритмів зважування за кутом та порогового розщеплення) на етапі підготовки геометрії і завершується оптимізованим рендерингом у вебсередовищі з використанням технологій WebGL. Інтеграція користувацьких шейдерів, правильна робота з TBN-матрицями та дотримання стандартів доступності (a11y) дають змогу створювати не лише візуально реалістичні, але й продуктивні та інклюзивні програмні рішення.

Список використаних джерел

1. Max N. Weights for computing vertex normals from facet normals. *Journal of Graphics Tools*. 1999. Vol. 4, № 2. P. 1–6.

2. Шикін Є. В., Боресков О. В. Комп’ютерна графіка. Полігональні моделі. Київ: ДіаСофт, 2001. 464 с.

3. Dirksen J. Learn Three.js: Program 3D animations and visualizations for the web with HTML5 and WebGL. 4th ed. Birmingham: Packt Publishing, 2023. 620 p.

4. glTF 2.0 Specification / Khronos Group. URL: https://registry.khronos.org/glTF/ (дата звернення: 11.05.2026).

5. Кадурін Д. О. Методи оптимізації тривимірних об’єктів для веб-додатків: навч. посіб. Харків: ХНУРЕ, 2022. 112 с.

УДК 004.7

Самоляк Д. В., здобувачка вищої освіти 3 курсу спеціальності 122 Комп’ютерні науки, Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ ГЛИБОКОГО НАВЧАННЯ (DEEP LEARNING) ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Прогнозування часових рядів є фундаментальним завданням у сфері інтелектуального аналізу даних, особливо в контексті стратегічного планування

та прийняття рішень. Сучасні інформаційні системи вимагають високої точності передбачень у динамічних середовищах, де класичні статистичні моделі (як-от ARIMA чи експоненціальне згладжування) часто не справляються з нелінійними залежностями та великими обсягами даних. Згідно з матеріалами Towards Data Science, саме перехід до архітектур глибокого навчання дав змогу значно підвищити якість прогнозів у фінансових та логістичних системах [1].

Однією з найбільш ефективних архітектур для роботи з послідовностями є рекурентні нейронні мережі (RNN), зокрема їх модифікація – довга короткострокова пам’ять (LSTM). Як зазначається у дослідженнях Machine Learning Mastery, мережі LSTM здатні вирішувати проблему зникаючого градієнта, що дає змогу їм «запам’ятовувати» довгострокові закономірності в даних, які були втрачені класичними методами [2]. Це критично важливо для аналізу ринкових циклів або сезонних коливань енергоспоживання.

Іншим перспективним напрямом, описаним у доповідях Google AI Blog, є використання одновимірних згорткових нейронних мереж (1D-CNN) для аналізу часових рядів. На відміну від рекурентних мереж, CNN дають змогу ефективно витягувати локальні ознаки та патерни, працюючи значно швидше на етапі навчання. Такий підхід часто застосовується у системах реального часу, де швидкість отримання прогнозу є критичним показником [3].

Сучасним стандартом у прогнозуванні послідовностей стають архітектури на основі механізмів уваги (Attention mechanism), зокрема трансформери (Transformers). За даними порталу Analytics Vidhya, використання трансформерів дає змогу моделі фокусуватися на найбільш значущих часових мітках у минулому, ігноруючи «інформаційний шум», що забезпечує кращу адаптивність системи до раптових змін середовища [4].

Альтернативні підходи охоплюють гібридні моделі, що поєднують розкладання рядів на компоненти (тренд, сезонність) із подальшою обробкою залишків нейронними мережами. На практиці вибір архітектури залежить від обсягу вибірки, частоти оновлення даних та вимог до точності. Отже, Deep Learning формує новий технологічний рівень у прогнозуванні, замінюючи жорсткі математичні моделі гнучкими адаптивними системами.

Як авторську пропозицію можна виділити використання ансамблевої архітектури, що поєднує LSTM для виявлення довгострокових трендів та 1D-CNN для аналізу короткострокових аномалій. Така комбінація дає змогу збалансувати глибину аналізу та чутливість моделі до різких стрибків даних. З практичного погляду, впровадження такої гібридної моделі в системи стратегічного управління допоможе знизити похибку прогнозування на 12–15 %, порівняно з однорідними нейромережами.

Отже, технології глибокого навчання є центральним інструментом у сучасному аналізі часових рядів. Розвиток методів Deep Learning забезпечує не лише високу точність передбачень, а й можливість автоматизації процесу прийняття стратегічних рішень у складних соціально-економічних системах.

Список використаних джерел

1. Deep Learning for Time Series Forecasting. *Towards Data Science.* URL: https://towardsdatascience.com/deep-learning-for-time-series-forecasting-36544520938

2. Understanding LSTM Networks. *Machine Learning Mastery*. URL: https://machinelearningmastery.com/long-short-term-memory-networks-deep-learning/

3. Convolutional Neural Networks for Time Series. *Google AI Blog*. URL: https://ai.googleblog.com/

4. Transformers in Time Series Analysis. *Analytics Vidhya*. URL: https://www.analyticsvidhya.com/blog/2022/01/attention-mechanism-for-time-series-forecasting/

УДК 004.01

Хуторна Д. В., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

МЕТОДИ ЯКОБІ ТА ЗЕЙДЕЛЯ У РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні рекомендаційні системи є важливим компонентом цифрових платформ, оскільки саме вони забезпечують персоналізацію контенту, підбір товарів, відео, музики та інформаційних матеріалів відповідно до інтересів користувача. У сучасних рекомендаційних системах обробляються великі обсяги даних про взаємодію користувачів з товарами, відео, музикою чи іншими інформаційними об'єктами. Оскільки значна частина таких даних залишається невідомою, у процесі обчислень доводиться працювати з розрідженими матрицями великої розмірності та неповними наборами спостережень. За таких умов важливого значення набувають чисельні методи розв'язання систем лінійних алгебраїчних рівнянь, які використовуються під час навчання та оновлення рекомендаційних моделей. [3; 5].

Найчастіше дані в рекомендаційних задачах подаються як матриця взаємодії користувачів і об'єктів. Більшість її елементів невідомі, тому класичне пряме відновлення повної матриці є неефективним. Натомість застосовують матричну факторизацію, коли початкова матриця апроксимується добутком двох матриць меншої розмірності. На практиці це перехід від аналізу мільйонів явних або неявних взаємодій до роботи з латентними факторами, що описують схожість користувачів та об'єктів у компактному вигляді. Така схема є основою багатьох промислових моделей рекомендацій, зокрема ALS-методу [3].

ALS (Alternating Least Squares) працює за принципом почергового уточнення параметрів: на одному кроці фіксуються фактори користувачів, на наступному – фактори об'єктів. Кожен із цих кроків зводиться до розв'язання системи лінійних алгебраїчних рівнянь. Саме тут виникає ключова обчислювальна проблема: такі системи потрібно розв'язувати багаторазово, а їх матриці є великими, розрідженими і часто погано пристосованими до прямих методів через високу вартість факторизації. Тому в промислових реалізаціях особливо цінними стають ітераційні схеми, які дають змогу отримувати прийнятне наближення без надмірних витрат часу та пам'яті [3; 4].

Сенс методу Якобі полягає в тому, що всі компоненти нового наближення обчислюються незалежно на основі попереднього вектора, тобто жодна нова компонента не впливає на інші в межах тієї самої ітерації. Це робить метод Якобі дуже зручним для паралельної реалізації, що є важливою перевагою для великих обчислювальних систем, кластерів і графічних прискорювачів. Однак саме така незалежність оновлень зменшує темп збіжності, метод часто потребує більшої кількості ітерацій, перш ніж наближення стане достатньо точним [1].

Метод Зейделя побудований інакше. Тут нові значення одразу використовуються в межах тієї ж ітерації, тому кожне наступне обчислення спирається вже не лише на попередній крок, а й на щойно уточнені компоненти. Це надає методу ефект «прискореного коригування» розв'язку. Для практичної чисельної математики такий механізм є дуже вигідним, оскільки зменшує кількість ітерацій до досягнення заданої точності. У рекомендаційних задачах це особливо важливо, бо зменшення числа ітерацій напряму скорочує час навчання моделі та знижує навантаження на обчислювальні ресурси [1].

Для практичних задач важливими є достатні умови типу діагонального домінування та характер ітераційної похибки. Якщо система має добре виражену діагональну перевагу, то обидва методи зазвичай збігаються, але метод Зейделя демонструє кращу динаміку зменшення похибки. У прикладному сенсі за однакової початковій точності Зейдель швидше наближається до стабільного розв'язку, тоді як Якобі виграє лише тоді, коли критичною є можливість паралельного виконання [1].

Похибка ітераційного методу визначається як різниця між поточним наближенням і точним розв'язком. Для рекомендаційних систем це не абстрактний показник, а реальний індикатор того, наскільки надійно модель відтворює приховані закономірності даних. Якщо похибка зменшується повільно, то система або працює довше, або потребує більших ресурсів для досягнення прийнятної якості. Якщо ж похибка спадає швидше, модель може бути навчена оперативніше, а отже, система стає придатнішою для практичного застосування у середовищах із великим потоком користувачів [1; 3].

Суттєво, що в recommendation systems ітераційні методи не є ізольованим математичним модулем. Вони безпосередньо впливають на те, наскільки швидко платформа оновлює профілі користувачів, перебудовує рекомендації та реагує на нові дані. Наприклад, у сервісах із високою частотою взаємодій затримка навіть у кілька секунд або хвилин може погіршити якість персоналізації. Тому перевага методу Зейделя полягає не лише у чисельній збіжності, а й у тому, що він краще відповідає вимозі швидкого практичного оновлення моделей у прикладному програмному забезпеченні [3; 5].

У теоретичному плані метод Якобі виглядає краще через свою просту структуру і придатність до паралельних обчислень. Проте для рекомендаційних систем, де ключовим є баланс між точністю й часом обчислення, метод Зейделя часто є вигіднішим. Його послідовне використання нових значень дає кращу локальну корекцію наближення, а отже, дає змогу досягти прийнятного результату за меншу кількість кроків. Саме тому в прикладному програмному забезпеченні, орієнтованому на обробку великих розріджених матриць, метод Зейделя має вищий практичний потенціал [1; 3].

Варто врахувати, що у сучасних роботах із graph collaborative filtering та ALS-подібних моделей ідеї ітераційного уточнення наближень зберігають свою актуальність, хоча самі алгоритми можуть бути модифіковані під специфіку графових даних або масштабованих систем. Це підтверджує, що методи Якобі та Зейделя залишаються базовими орієнтирами для аналізу ефективності чисельних процедур у штучному інтелекті та рекомендаційних сервісах [2; 4].

Отже, системи лінійних алгебраїчних рівнянь є фундаментом математичного моделювання сучасних recommendation systems. Метод Якобі доцільний там, де важлива паралельність обчислень, а метод Зейделя – там, де пріоритетом є швидкість збіжності та зменшення обчислювальної похибки. Для прикладного програмного забезпечення це означає прямий виграш у швидкодії, що особливо цінно для великих AI-сервісів із високим навантаженням. Отже, метод Зейделя треба вважати більш ефективним інструментом для підвищення оперативності рекомендаційних систем у середовищі Big Data.

Список використаних джерел

1. Khrapov P., Volkov N. Comparative analysis of Jacobi and Gauss-Seidel iterative methods. *International Journal of Open Information Technologies.* 2024. Vol. 12, iss. 2. P. 23–34.

2. On Manipulating Signals of User-Item Graph: A Jacobi Polynomial-based Graph Collaborative Filtering / J. Guo, L. Du, X. Chen et al. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* New York: ACM, 2023. P. 3329–3340. DOI: 10.1145/3580305.3599450.

3. Revisiting the Performance of iALS on Item Recommendation Benchmarks / S. Rendle, W. Krichene, L. Zhang, Y. Koren. *Proceedings of the 16th ACM Conference on Recommender Systems.* New York: ACM, 2022. P. 427–435. DOI: 10.1145/3523227.3548486.

4. Core-elements Subsampling for Alternating Least Squares / D. Xue, M. Li, C. Meng, J. Zhang. *arXiv preprint.* arXiv:2509.18024. 2025.

5. Дослідження методів побудови рекомендаційних систем для розв'язання задачі вибору найбільш релевантного відео при створенні віртуальних арт-композицій / А. Кулягін, В. Нарожний, В. Ткачов, Г. Кучук. *Системи управління, навігації та зв'язку.* 2022. Т. 4, № 72. С. 94–99. DOI: 10.26906/SUNZ.2022.4.094.

УДК 004.85

Ярошенко Б. М., здобувач вищої освіти 3 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

ОПТИМІЗАЦІЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ЗА ДОПОМОГОЮ РЕГУЛЯРИЗАЦІЇ МЕТОДАМИ L1 І L2

Донецький національний університет імені Василя Стуса, м. Вінниця

Однією з ключових проблем у машинному навчанні є перенавчання (overfitting) – явище, за якого модель надмірно адаптується до тренувальних даних, включно з їх шумом та випадковими відхиленнями, і втрачає здатність до

узагальнення на нових даних. На тренувальній вибірці така модель демонструє високу точність, однак на тестовій – суттєво гіршу [1]. Для боротьби з перенавчанням широко застосовуються методи регуляризації – математично обґрунтовані підходи, що обмежують складність моделі в процесі навчання.

Перенавчання виникає, коли модель є надмірно складною відносно обсягу навчальних даних: вона фактично запам'ятовує конкретні приклади замість виявлення загальних закономірностей. Протилежним явищем є недонавчання (underfitting), коли модель занадто проста і не вловлює базових залежностей у даних. Оптимальна модель балансує між цими двома крайнощами [1; 2].

На практиці перенавчання особливо гостро проявляється за умови малого обсягу навчальних даних, великої кількості параметрів моделі або в задачах із зашумленими даними. Виявити перенавчання дає змогу порівняння точності моделі на тренувальній і валідаційній вибірках: суттєва розбіжність між ними є надійним індикатором проблеми [1].

Регуляризація L2 (Ridge) додає до функції втрат штраф у вигляді суми квадратів усіх коефіцієнтів моделі: $L = L_0 + \lambda \cdot \sum {w_i}^2$. Параметр λ контролює інтенсивність штрафу. Цей підхід змушує модель утримувати малі значення коефіцієнтів, не допускаючи домінування жодної ознаки. Важливо, що L2 не обнуляє коефіцієнти – усі ознаки залишаються в моделі з меншим впливом. Метод є особливо ефективним за мультиколінеарності та коли більшість ознак мають прогностичну цінність [3].

Регуляризація L1 (Lasso) відрізняється формою штрафу: до функції втрат додається сума абсолютних значень коефіцієнтів: $L = L_0 + \lambda \cdot \Sigma|w_i|$. Геометрично це призводить до того, що оптимум часто досягається в точках, де частина коефіцієнтів дорівнює нулю, – тобто L1 здійснює автоматичний відбір ознак, виключаючи незначущі змінні з моделі. Це робить Lasso особливо корисним у задачах з великою кількістю потенційних ознак, більшість з яких є нерелевантними [3].

Вибір між L1 і L2 залежить від характеру задачі та природи даних. L2 переважніше обирають, коли більшість ознак є корисними або присутня мультиколінеарність. L1 – коли потрібна інтерпретована розріджена модель або кількість ознак значно перевищує кількість спостережень. Існує також комбінований метод Elastic Net, що поєднує штрафи L1 і L2: він одночасно здійснює відбір ознак і стабільно працює під час кореляції між ними [4].

Обидва методи широко застосовуються в медичній діагностиці, фінансовому прогнозуванні та обробці природної мови. В усіх випадках оптимальне значення параметра λ підбирається методом крос-валідації: за $\lambda = 0$ регуляризація відсутня, за великих значень – модель стає надмірно обмеженою [2].

Отже, перенавчання є фундаментальною проблемою машинного навчання, що безпосередньо впливає на практичну придатність моделей. Регуляризація L1 і L2 є ефективними математично обґрунтованими інструментами боротьби з цим явищем. L2 зберігає всі ознаки, зменшуючи їх вплив; L1 виконує автоматичний відбір ознак, обнуляючи незначущі коефіцієнти. Розуміння природи цих методів і вміння обирати між ними є необхідною компетенцією для практичного застосування алгоритмів статистичного і машинного навчання.

Список використаних джерел

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 p.
2. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Springer, 2009. 745 p.
3. Tibshirani R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society.* 1996. Vol. 58, № 1. P. 267–288.
4. Zou H., Hastie T. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society.* 2005. Vol. 67, № 2. P. 301–320.

УДК 004.932

Федоришин Б. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук, старший викладач кафедри інформаційних технологій

ЕВОЛЮЦІЯ МЕТОДІВ АВТОМАТИЗАЦІЇ ПЕРЕКЛАДУ УКРАЇНСЬКОЇ ЖЕСТОВОЇ МОВИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Україна наразі перебуває в особливо критичному становищі щодо поширеності порушень слуху. За даними ГО «Відчуй», яка з 2011 р. надає допомогу українцям із порушеннями слуху, в Україні налічується понад 2,1 мільйона людей із вадами слуху різного ступеня, з яких приблизно 230 тисяч використовують жестову мову як основний або один з основних засобів комунікації [1]. З початком повномасштабного вторгнення Росії у лютому 2022 р. ситуація суттєво погіршилась: кількість людей, які повністю або частково втратили слух, зросла приблизно на 20 % [1]. Особливо вразливою групою є військовослужбовці: за оцінками фахівців організації, кожен другий український військовий втрачає слух унаслідок бойових дій і потребує слухопротезування та подальшої реабілітації. Погіршення слуху фіксують також у цивільного населення на деокупованих територіях внаслідок акустичних баротравм від вибухів, тривалого перебування в укриттях та обмеженого доступу до медичної допомоги. Це зумовлює стійкий і зростаючий комунікаційний бар'єр між нечуючими та чуючими громадянами, суттєво обмежуючи доступ перших до базових соціальних послуг. Українська жестова мова (УЖМ) функціонує як повноцінна самостійна лінгвістична система зі складною просторово-часовою граматикою. Попри прогрес у галузі штучного інтелекту, фундаментальною проблемою залишається критична відсутність дієвих автоматизованих систем, здатних розпізнавати динамічні жести УЖМ безпосередньо з відеосигналу. Наявні вітчизняні рішення здебільшого обмежуються функціоналом статичного дактилю або текстового перекладу заздалегідь анотованих глос. Метою роботи є систематизація еволюції методів автоматизації сурдоперекладу та обґрунтування доцільності застосування архітектур LSTM/GRU у синергії з інструментарієм MediaPipe для вирішення цього завдання.

Перші спроби автоматизації сурдоперекладу базувалися на детермінованих символьних та онтологічних підходах. Зокрема, у працях О. Лозинської розроблено граматично доповнену онтологію для встановлення структурних відповідностей між текстами УСМ та глосами УЖМ [2]. Проте такі системи оперують виключно текстовими анотаціями, ігноруючи візуальну природу жестів. Наступним кроком стало застосування статистичного машинного перекладу. За допомогою моделей IBM та алгоритму очікування-максимізації вирішувалася проблема відмінного синтаксичного порядку знаків у жестовій та словесній мовах [3]. Перехід до аналізу неструктурованого відеосигналу відбувся із залученням класичного комп'ютерного зору, операторів Canny, дескрипторів HOG у поєднанні з прихованими марковськими моделями, що детально розглянуто в дослідженнях Т. Басюка та А. Василюка [4]. Однак ці алгоритми виявилися критично чутливими до умов зйомки та вимагали складного ручного проєктування ознак. Сучасний етап характеризується переходом до глибокого навчання: інтеграція MediaPipe із рекурентними архітектурами LSTM/GRU забезпечує автоматичне видобування просторово-часових закономірностей [5]. Попри прогрес, жодна з наявних систем не пропонує завершеного рішення для відеорозпізнавання динамічної лексики УЖМ у режимі реального часу.

Метою дослідження є проєктування та практична розробка програмного комплексу для розпізнавання і класифікації динамічних жестів УЖМ у режимі реального часу. Система поєднує MediaPipe Holistic для екстракції просторових ознак та архітектуру LSTM для аналізу часових рядів. Для досягнення мети виконано такі завдання: проаналізовано еволюцію підходів до автоматизації перекладу УЖМ; сформовано датасет відеозаписів динамічних жестів у різних умовах зйомки; розроблено та навчено LSTM-модель для класифікації 10 жестів.

Розроблений програмний комплекс реалізує п'ятирівневий конвеєр обробки даних. На першому рівні засобами OpenCV здійснюється захоплення відеопотоку з вебкамери. На другому рівні бібліотека MediaPipe Holistic виконує покадрове виділення координат ключових точок тіла та рук, загалом 258 числових ознак (поза тіла: $33 \times 4 = 132$; ліва рука: $21 \times 3 = 63$; права рука: $21 \times 3 = 63$). Точки обличчя навмисно виключено з вектора ознак з двох причин: по-перше, їх детекція є нестабільною за різних умов зйомки; по-друге, виключення $468 \times 3 = 1\,404$ ознак суттєво підвищило продуктивність системи. На третьому рівні вектори ознак накопичуються у ковзне вікно тривалістю 30 кадрів, що відповідає середній тривалості одного динамічного жесту. Четвертий рівень реалізує інференс навченої LSTM-моделі: тризарова архітектура з 64-128-64 нейронами, шарами Dropout ($p = 0.2$) та Softmax-виходом. П'ятий рівень здійснює стабілізоване відображення результату: жест виводиться лише за умови його підтвердження у 10 послідовних передбаченнях з рівнем впевненості понад 80 %.

Датасет сформовано з власноруч зібраних відеозаписів 10 жестів. Для кожного жеста залучено двох акторів, кожен із яких виконував жест у двох форматах запису: вертикальному (смартфон) та горизонтальному (фронтальна камера). З кожного відео методом ковзного вікна виділено по 15 послідовностей, що забезпечило репрезентативне охоплення варіацій виконання. Дані роз-

поділено у співвідношенні 80/10/10 для навчальної, валідаційної та тестової вибірок відповідно. Для обґрунтування вибору архітектури проведено порівняльний експеримент трьох рекурентних моделей на однаковому датасеті (50 епох).

Таблиця 1 – Порівняння рекурентних архітектур

Архітектура	**Accuracy (50 еп.)**	**Час навчання**	**Обрано**
SimpleRNN	26.47 %	29 с	О
GRU	82.35 %	62 с	О
LSTM (200 еп.)	94.12 %	~8 хв	✓

Фінальна LSTM-модель навчалася протягом 200 епох із callbacks EarlyStopping (patience = 25) та ModelCheckpoint. За результатами тестування на відкладеній вибірці досягнуто точність 94.12 % (Loss = 0.1977). Водночас у процесі практичного тестування в режимі реального часу виявлено характерне обмеження: за продуктивності системи 10–15 FPS тимчасове вікно з 30 кадрів охоплює 2–3 секунди реального часу, тоді як навчальні відео оброблялися за 25–30 FPS. Це розходження між частотою кадрів під час навчання та інференсу призводить до того, що модель отримує «уповільнені» у часі послідовності, що знижує точність розпізнавання окремих жестів у реальних умовах.

Проведений аналіз засвідчує чіткий хронологічний перехід від символьних підходів до нейромережевих методів обробки відеоданих. Кожен еволюційний крок усував обмеження попередника: символьні методи замінено статистичними, ті – комп’ютерним зором, а той – архітектурами глибокого навчання. Розроблений комплекс є першою практичною реалізацією для УЖМ, що вирішує задачу на рівні безпосереднього відеорозпізнавання без попередньої текстової анотації, досягаючи точності 94.12 % на тестовій вибірці. Виявлені технічні обмеження, невідповідність FPS під час навчання та інференсу, а також плутанина між візуально схожими жестами. Ці проблеми окреслюють напрями подальшого вдосконалення: адаптивна нормалізація часових послідовностей, розширення датасету та інтеграція немануальних компонентів.

Список використаних джерел

1. Герасимчук Д. «Чуєш? Плюс. Плюс» – Як відновлюють слух українським військовим / *ГО «Відчуй» для The Village Україна.* 2024. URL: https://vidchui.org/2024/02/09/chuyesh-plyus-plyus-yak-vidnovlyuyut-sluh-ukrayinskym-vijskovym-rozpovidayemo-dlya-the-village-ukrayina/

2. Лозинська О. В. Онтологічний підхід до автоматизованого перекладу між українською словесною та жестовою мовами. *Штучний інтелект.* 2016. № 3–4. С. 105–113.

3. Лозинська О. В. Статистичний машинний переклад між українською словесною та жестовою мовами. *Проблеми програмування.* 2024. № 1. С. 45–54.

4. Басюк Т. М., Василюк А. С. Огляд методів розпізнавання жестів мови в системах комп’ютерного зору. *Вісник НТУУ «КПІ».* 2024. Вип. 78.

5. Зайцева М. О. Розпізнавання жестів на основі MediaPipe та рекурентних нейронних мереж. *Молодий вчений.* 2025. № 2. С. 112–118.

СЕКЦІЯ 4
ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ТА ПРИЙНЯТТЯ РІШЕНЬ

УДК 004.852:004

Балюра Б. П., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Ніколюк П. К., д-р фіз.-мат. наук, професор, професор кафедри інформаційних технологій

АЛГОРИТМИ ТА ПРОГРАМНІ ЗАСОБИ ВИЯВЛЕННЯ АНОМАЛІЙ У ДАНИХ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Виявлення аномалій *(anomaly detection)* є однією з ключових задач інтелектуального аналізу даних, що набуває особливого значення в умовах стрімкого зростання обсягів інформації. За оцінками дослідників, щорічні світові збитки від карткового шахрайства перевищують 32 млрд доларів [1], а кібератаки коштують підприємствам у середньому 4,5 млн доларів кожна [2]. Традиційні підходи на основі вручну написаних правил не справляються з масштабом та динамікою сучасних потоків даних, що обумовлює актуальність розробки інтелектуальних систем автоматичного виявлення відхилень.

Аналіз сучасних досліджень засвідчує, що серед алгоритмів машинного навчання без учителя для задач anomaly detection найбільшого поширення набули: Isolation Forest [3], Local Outlier Factor (LOF) [4] та One-Class SVM. Автоенкодери на основі глибинного навчання демонструють перспективні результати, однак вимагають значних обчислювальних ресурсів та залежностей від спеціалізованих фреймворків [5]. Відсутні комплексні програмні засоби, що об'єднують кілька алгоритмів в єдиному середовищі з AI-інтерпретацією результатів для нетехнічних користувачів. Новизна роботи полягає у розробці інтегрованої системи, що поєднує детекцію аномалій з AI-інтерпретацією кожного виявленого відхилення, що відрізняє її від наявних інструментів, де пояснення результатів покладається на користувача.

Метою дослідження є розробка програмного засобу для виявлення аномалій у даних на основі алгоритмів машинного навчання з інтеграцією штучного інтелекту для автоматичної інтерпретації результатів. Для досягнення мети розв'язано такі задачі: проведено порівняльний аналіз алгоритмів Isolation Forest, LOF, One-Class SVM та Autoencoder за метриками Precision, Recall, F1-score та ROC-AUC; розроблено багатомодульну архітектуру програмної системи; проведено експериментальне дослідження на реальному датасеті; інтегровано AI-асистент для пояснення виявлених аномалій природною мовою.

Розроблений програмний засіб **AnomalyAI** реалізовано мовою Python із використанням бібліотек scikit-learn (алгоритми ML), Plotly (інтерактивна візуалізація) та PySide6 (графічний інтерфейс). Архітектуру побудовано за чотиришаровим принципом: *core* – алгоритми детекції, *services* – обробка даних та AI, *viz* – візуалізація, *ui* – графічний інтерфейс. Система підтримує п'ять прикладних доменів: фінанси, IoT-датчики, кібербезпека, охорона здоров'я та e-commerce.

Autoencoder реалізовано без залежності від TensorFlow – на основі методу головних компонент (PCA). Кодер виконує проєкцію у простір зниженої розмірності, декодер відновлює вихідні дані через зворотне PCA-перетворення. Аномалії визначаються як записи з максимальною середньоквадратичною помилкою реконструкції, що забезпечує доступність алгоритму у будь-якому середовищі Python без додаткових залежностей.

Для візуалізації результатів реалізовано дев'ять типів інтерактивних графіків: PCA/t-SNE Embedding – відображення кластерів аномалій у двовимірному просторі; Time Series – аналіз тимчасових патернів аномального score; Feature Importance – ранжування ознак за їх внеском у рішення моделі; Score Distribution; Risk Heatmap; ROC Curves; Model Comparison; Confusion Matrix та Data Table. AI Copilot на базі Gemini API пояснює причини класифікації кожної аномалії природною мовою у форматі: «показник X на +7,3σ вище норми».

Експериментальне дослідження проведено на датасеті Credit Card Fraud Detection (Kaggle) [1], що містить 284 807 реальних банківських транзакцій, з яких 492 є шахрайськими (0,17 %). Результати порівняльного аналізу алгоритмів наведено у табл. 1.

Таблиця 1 – Порівняння алгоритмів виявлення аномалій на датасеті Credit Card Fraud Detection

Алгоритм	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
Isolation Forest	0,047	0,778	0,088	0,909
One-Class SVM	0,051	0,842	0,096	0,947
LOF	0,006	0,143	0,013	0,492
Autoencoder (PCA)	0,038	0,694	0,072	0,871

One-Class SVM продемонстрував найвищий ROC-AUC = 0,947 та Recall = 0,842, виявляючи 84 % реального шахрайства. Isolation Forest є оптимальним за співвідношенням якість / швидкість – ROC-AUC = 0,909 за умови часу обробки 284 807 записів приблизно 30 секунд завдяки оптимізованому субсемплінгу (max_samples = 256 для великих датасетів). LOF показав низьку ефективність (AUC = 0,492): аномалії у просторі 30 PCA-ознак датасету є глобальними, що суперечить локальному принципу роботи алгоритму. Низькі значення Precision для всіх моделей пояснюються суттєвим дисбалансом класів – реальна частка шахрайства (0,17 %) значно менша за встановлений параметр contamination (4 %). Ключовою ознакою для детекції визначено компонент V1 з нормалізованою важливістю 1,0.

Висновки. Розроблений програмний засіб AnomalyAI об'єднує чотири алгоритми виявлення аномалій, дев'ять типів інтерактивних візуалізацій та AI-інтерпретацію результатів у єдиному desktop-додатку. Систему успішно протестовано на реальному датасеті з 284 807 записів – досягнуто ROC-AUC = 0,947 (One-Class SVM) за умови часу обробки до 30 секунд. Практична цінність розробки полягає у можливості застосування у банківських системах, промисловому IoT, кібербезпеці та медичній діагностиці без необхідності спеціальних технічних знань від кінцевого користувача. Перспективами подальшого розвитку

є інтеграція підтримки потокових даних у реальному часі та розширення набору алгоритмів глибинного навчання.

Список використаних джерел

1. Learned lessons in credit card fraud detection from a practitioner perspective / A. Dal Pozzolo, O. Caelen, Y.-A. le Borgne, S. Waterschoot, G. Bontempi. *Expert Systems with Applications.* 2014. Vol. 41(10). P. 4915–4928.

2. IBM Security. Cost of a Data Breach Report 2023. *IBM Corporation.* URL: https://www.ibm.com/reports/data-breach (дата звернення: 15.04.2026).

3. Liu F. T., Ting K. M., Zhou Z.-H. Isolation Forest. *Proceedings of the 8th IEEE International Conference on Data Mining.* Pisa, 2008. P. 413–422.

4. LOF: Identifying Density-Based Local Outliers. M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander. *Proceedings of ACM SIGMOD International Conference on Management of Data.* Dallas, 2000. P. 93–104.

5. Chalapathy R., Chawla S. Deep Learning for Anomaly Detection: A Survey. *arXiv preprint.* arXiv:1901.03407. 2019. 50 p.

УДК 004.9:004.65:613.2

Бевзюк А. Ю., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Антонов Ю. С., канд. фіз.-мат. наук, доцент, доцент кафедри інформаційних технологій

ПРОЄКТУВАННЯ БАЗИ ДАНИХ СИСТЕМИ ПЕРСОНАЛЬНОГО КОНТРОЛЮ ХАРЧУВАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах дедалі більше уваги приділяється підтримці здорового способу життя, особливо контролю харчування та забезпеченню збалансованості раціону. Неправильний режим прийому їжі, надмірне споживання калорій та відсутність індивідуального підходу до формування раціону можуть спричиняти розвиток хронічних захворювань і погіршення загального стану здоров'я [1]. У зв'язку з цим інформаційні технології відіграють важливу роль у підтримці здорового способу життя, забезпечуючи автоматизацію збору та аналізу даних про харчування. Зокрема, актуальною є розробка інформаційних систем, що здійснюють моніторинг харчування та надають користувачам персоналізовані рекомендації.

Ключовим компонентом таких систем є база даних, яка забезпечує зберігання, обробку та аналіз даних користувачів, їхніх харчових звичок, обмежень та цілей. Архітектура бази даних визначає продуктивність системи, її масштабованість, а також ефективність виконання аналітичних запитів [2].

Процес моделювання бази даних передбачає розробку концептуальної моделі предметної області, що базується на визначенні основних сутностей, їх атрибутів та взаємозв'язків для подальшого об'єктно-реляційного відображення [3].

На основі аналізу предметної області було сформовано перелік ключових об’єктів, які відображають процеси обліку харчування, відстеження стану користувача та формування рекомендацій.

Узагальнений перелік сутностей та їх основних атрибутів наведено в табл. 1.

Таблиця 1 – Основні об’єкти інформаційної моделі та їх властивості

Об’єкт (процес)	**Властивості**
Користувач	Ім’я, дата народження, стать, зріст, рівень фізичної активності, початкова вага
Прогрес користувача	Вага, дата вимірювання
Ціль користувача	Тип (підтримка ваги / набір маси / схуднення), цільова вага
Обмеження та алергени	Назва, тип обмеження (алергія / дієта / хвороба), рівень критичності
Продукти	Назва, категорія, калорії (ккал на 100 г / 100 мл), БЖВ, цукор, вітаміни (A, B, C), клітковина, алергени, одиниця вимірювання (г, мл, шт.)
Страви	Назва, опис, розмір порції, категорія, БЖВ, калорії
Склад страви	Продукт, кількість, одиниця вимірювання (г, мл, шт.)
Денна норма	Дата встановлення, калорії, БЖВ, вітаміни, клітковина, норма води
Прийоми їжі	Тип, дата та час, список страв / продуктів, фактична вага
Вода	Дата, кількість (мл)
Рекомендації	Тип, опис, рекомендовані продукти / страви

Наведена таблиця описує концептуальний рівень моделі даних і слугує основою для подальшого логічного проєктування бази даних, зокрема нормалізації структури та побудови схеми відношень [4]. Модель відображає основні інформаційні потоки системи та дає змогу визначити ключові взаємозв’язки між сутностями. Важливою особливістю моделі є врахування як статичних даних, так і динамічних процесів, зокрема обліку прийомів їжі та змін фізичних показників користувача.

До того ж модель передбачає можливість розширення функціоналу системи за рахунок додавання нових типів даних, як-от додаткові нутрієнти, категорії продуктів або види рекомендацій, що забезпечує її гнучкість та адаптивність до змін вимог.

На основі сформованої моделі було розроблено логічну схему бази даних, яка відображає структуру таблиць, їх атрибути, а також зв’язки між ними. Під час цього було проведено нормалізацію даних з метою усунення надлишковості та забезпечення цілісності даних [5].

Фрагмент логічної схеми бази даних, що ілюструє підсистему обліку прийомів їжі та структуру страв, наведено на рис. 1.

На представленому фрагменті схеми відображено взаємозв’язок між прийомами їжі, продуктами та стравами. Центральним елементом даної підсистеми є таблиця MealItem, яка реалізує універсальний механізм зберігання як окремих продуктів, так і складених страв у межах одного прийому їжі. Склад страв моделюється за допомогою зв’язувальної таблиці DishIngredients, що дає змогу реалізувати зв’язок типу «багато-до-багатьох» між стравами та продуктами.

Такий підхід забезпечує деталізоване представлення харчування користувача та створює основу для подальшого аналізу раціону. Загалом база даних складається з 29 таблиць.

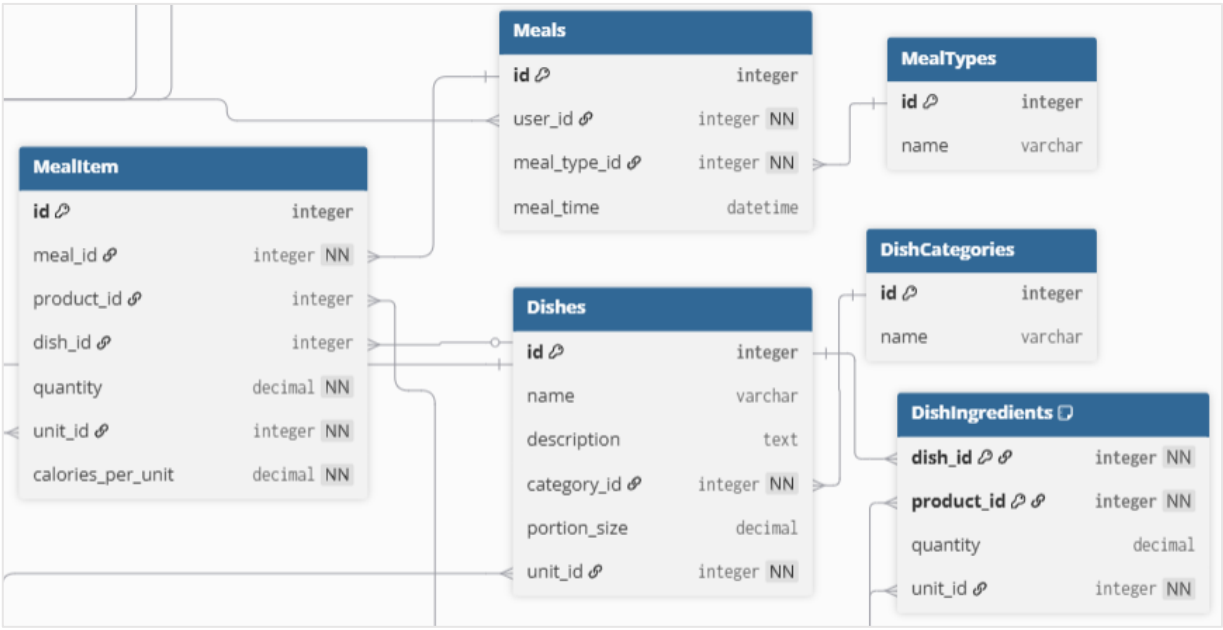


Рисунок 4 – Фрагмент логічної схеми бази даних системи

Розроблена логічна структура бази даних забезпечує цілісність, несуперечливість та ефективне зберігання даних. Використання принципів нормалізації дало змогу зменшити надлишковість інформації та підвищити гнучкість системи, що є важливим для реалізації персоналізованого підходу до контролю харчування.

Список використаних джерел

1. Stephenson T. J., Sanctuary M., Passerrello C. Human Nutrition: Science for Healthy Living: 2024 Release ISE. 4th ed. McGraw-Hill Education, 2024. 832 p.

2. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, Incorporated, 2017.

3. Smith J. P. Entity Framework Core in Action, Second Edition. Manning Publications Co. LLC, 2021. 624 p.

4. Hernandez M. J. Database Design for Mere Mortals: 25th Anniversary Edition. Pearson Education, Limited, 2020.

5. Fundamentals of database systems / ed. by N. Sham. 6th ed. Boston: Addison-Wesley, 2011. 1172 p.

УДК 004.8

Богач Т. О., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Фриз І. В., канд. фіз.-мат. наук, старший викладач кафедри інформаційних технологій

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ВИБОРУ КРАЇНИ ПЕРЕБУВАННЯ УКРАЇНСЬКИХ МІГРАНТІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Переміщення населення є природним процесом, який супроводжує людство протягом усієї історії. Сьогодні міграція набуває дедалі більшого масштабу та стає однією з найактуальніших проблем як для економічно розвинених держав, так і для країн, що розвиваються. У сучасних умовах зростання масштабів міжнародної міграції українські громадяни стикаються з необхідністю вибору країни для тимчасового або постійного проживання, працевлаштування чи навчання [1]. Прийняття такого рішення ускладнюється великою кількістю факторів, серед яких рівень заробітної плати, умови праці, рівень безпеки, соціальний захист, вартість життя, мовне середовище, доступність медичних та освітніх послуг, міграційна політика держави та інші критерії.

Аналіз зазначених факторів вручну є складним і потребує значних часових витрат, що зумовлює необхідність створення автоматизованої інтелектуальної системи підтримки прийняття рішень для вибору оптимальної країни перебування українських мігрантів на основі аналізу соціально-економічних, правових та демографічних показників [2]. Штучний інтелект у таких системах використовується для оцінювання та порівняння країн за визначеними критеріями з метою підвищення ефективності процесу прийняття рішень щодо міграції та адаптації українських громадян за кордоном.

Метою дослідження є створення системи, яка забезпечуватиме збір, обробку та аналіз даних про потенційні країни перебування, а також формування рекомендацій для користувача відповідно до його індивідуальних потреб і пріоритетів. Для цього необхідно реалізувати механізми багатокритеріального аналізу, оцінювання альтернатив та ранжування країн за визначеними показниками. Інтелектуальна система повинна враховувати персональні параметри користувача, як-от професія, рівень освіти, знання іноземних мов, сімейний стан, фінансові можливості та мета міграції. На основі введених даних система визначає найбільш придатні країни для перебування та надавати обґрунтовані рекомендації щодо вибору оптимального варіанта. Критерії для оцінювання країни для міграції: правовий статус, фінансова підтримка, працевлаштування та економіка, житло та інтеграція.

Оскільки вибір країни залежить від багатьох факторів (безпека, мова, фінанси, соціум), використовуються класичні інтелектуальні методи:

– метод аналізу ієрархій (AHP – Analytic Hierarchy Process), який дає змогу користувачеві попарно порівнювати критерії (наприклад: «Що важливіше: висока

зарплата чи простота вивчення мови?»), водночас ШІ розраховує вектори пріоритетів і перевіряє матрицю на узгодженість;

– метод TOPSIS, що дає змогу знаходити країну, яка є геометрично найближчою до «ідеального рішення» і найвіддаленішою від «найгіршого сценарію».

Алгоритм прийняття рішення полягає в таких кроках:

1) оцінка ресурсів, тобто розраховується стартовий капітал (наприклад, сума в еквіваленті $2 000–5 000 для базової адаптації);

2) аналіз рівня володіння мовою країни або міжнародною (англійською);

3) порівняння ринку праці на основі аналізу сайтів з пошуку роботи в цільовій країні (наприклад, LinkedIn або Indeed);

4) оцінка клімату та відстані до України з метою оцінки зручності та вартості логістики для підтримки зв’язку з домом.

Отже, інтелектуальна система надає список країн та міст, що може обрати людина під час планування процесу міграції. Це дає змогу обрати найбільш оптимальну країну або місто для переїзду. Список ранжується в порядку спадання від найкращої пропозиції до найгіршої.

За результатами роботи розроблено інтелектуальну систему прийняття рішень, що дає змогу на основі введеної інформації отримати перелік країн перебування українських мігрантів з урахуванням особливостей та потреб користувача.

Список використаних джерел

1. Як переїхати жити в Європу: способи імміграції у 2026 р. *EUdoc.* URL: https://eudoc.com.ua/immigracziya-v-yevropu/#close (дата звернення: 01.05.2026).

2. Information for Refugees, Asylum-seekers and Stateless People | UNHCR. *UNHCR help.unhcr.org.* URL: https://help.unhcr.org/ (дата звернення: 01.05.2026).

УДК 004.01

Грищишен О. Ю., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп’ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ЗАДАЧ КЛАСИФІКАЦІЇ ПАТЕРНІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. В епоху глобальної цифровізації та безперервного накопичення великих обсягів даних виникає гостра потреба в ефективних інструментах для їх обробки та аналізу. Серед ключових завдань сучасних інформаційних систем особливе місце займає розпізнавання та класифікація патернів у зображеннях, багатовимірних масивах даних та сигналах.

Актуальність. Застосування класичних алгоритмів машинного зору та машинного навчання часто не дає змоги досягти необхідного рівня точності

під час роботи зі складними та неструктурованими даними. У цьому контексті моделі глибокого навчання (Deep Learning) демонструють передові результати завдяки здатності самостійно виділяти ієрархічні ознаки. Тому розробка та оптимізація програмних рішень на їх основі є вкрай актуальним завданням для багатьох сфер: від медичної діагностики до промислового моніторингу.

Аналіз останніх досліджень. Сучасні дослідження у сфері штучного інтелекту значною мірою спираються на використання згорткових (CNN), рекурентних (RNN) нейронних мереж та архітектури Transformer. Провідні науковці довели беззаперечну ефективність таких архітектур для класифікації складних патернів [1]. Водночас питання архітектурної та апаратної оптимізації цих моделей під час програмної реалізації для конкретних прикладних задач залишається відкритим і потребує подальших розвідок [2].

Постановка задачі. Для досягнення поставленої мети необхідно: проаналізувати наявні архітектури нейромереж, визначити оптимальні метрики та функції втрат для класифікації, а також спроєктувати структуру програмного модуля, здатного результативно обробляти вхідні набори даних.

Мета роботи полягає у визначенні архітектурних особливостей та перспектив розробки ефективного програмного модуля на базі моделей глибокого навчання для вирішення завдань автоматичної класифікації патернів.

Викладення основного матеріалу. Основою програмної реалізації модуля є побудова багатошарової нейромережевої архітектури, що включає вхідний шар, набір прихованих шарів (згорткових, агрегуючих, повнозв'язних) та вихідний шар для безпосередньої класифікації. На відміну від традиційних підходів, глибоке навчання автоматизує процес вилучення ознак (feature extraction) з сирих даних.

Критично важливим етапом навчання моделі є процес оптимізації ваг шляхом мінімізації похибки. Для задач багатокласової класифікації патернів найефективнішим підходом є використання функції втрат перехресної ентропії (Cross-Entropy Loss), яка оцінює розбіжність між передбаченим моделлю розподілом ймовірностей та фактичним:

$$L = -\sum_{i=1}^{C} y_i log(\hat{y}_i),$$

де C – загальна кількість класів патернів;

y_i – бінарний індикатор (0 або 1), що позначає, чи є клас i правильним для поточного спостереження;

$\hat{y}_i$ – передбачена нейромережею ймовірність того, що об'єкт належить до класу i.

Програмна реалізація такого модуля передбачає використання сучасних фреймворків (TensorFlow, PyTorch), які підтримують тензорні обчислення та дають змогу ефективно розпаралелювати їх на графічних процесорах (GPU). Розроблена система повинна включати кілька підсистем: модуль попередньої обробки та аугментації даних для підвищення стійкості моделі, безпосередньо блок інференсу (класифікації) та програмний інтерфейс (API) для взаємодії з

користувачем або іншими системами [3]. Впровадження такого підходу дасть змогу масштабувати програмне рішення та адаптувати його під різні класи вхідних об’єктів без суттєвої зміни вихідного коду алгоритму.

Висновки. Дослідження та подальша програмна реалізація моделей глибокого навчання відкривають широкі перспективи для автоматизації рутинних процесів ідентифікації та класифікації патернів. Використання багатошарових нейромереж у поєднанні з функцією перехресної ентропії дає змогу створити гнучкий і високоточний програмний модуль. Подальші дослідження будуть спрямовані безпосередньо на практичну реалізацію розробленої архітектури та її тестування на реальних датасетах для підтвердження обчислювальної ефективності.

Список використаних джерел

1. LeCun Y., Bengio Y., Hinton G. Deep learning. Nature. 2015. Vol. 521, № 7553. P. 436–444.
2. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 p.
3. Chollet F. Deep Learning with Python. Manning Publications, 2017. 384 p.

УДК 004.65:658.82

Зигарь Д. Д., здобувач вищої освіти 4 курсу спеціальності 122 Комп’ютерні науки,
Баценко Т. В., канд. техн. наук, доцент, доцент кафедри інформаційних технологій

АНАЛІЗ ДАНИХ У СИСТЕМАХ ЛОЯЛЬНОСТІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасний IT-сектор у ритейлі та e-commerce перейшов від простої фіксації продажів до глибокого аналізу поведінки споживачів. Системи лояльності сьогодні – це не лише нарахування балів, а потужне джерело даних для сегментації аудиторії та прогнозування прибутку. Для розробника це означає перехід від звичайного CRUD (створення, читання, оновлення, видалення даних) до впровадження аналітичних алгоритмів, що допомагають бізнесу зростати.

Одним із найбільш ефективних методів аналізу є модель RFM (Recency, Frequency, Monetary). Вона базується на трьох простих метриках: як давно була покупка, як часто клієнт повертається та скільки грошей він витрачає. Використання мови Python дає змогу автоматизувати цей процес, перетворюючи тисячі рядків із бази даних на зрозумілі графіки, що вказують на «вузькі місця» у взаємодії з клієнтами.

Нижче наведено приклад оптимізованого коду, який імітує роботу з базою даних, розраховує ключові показники та візуалізує стан клієнтської бази для прийняття бізнес-рішень:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
```

```
data = [
    [101, datetime.now() - timedelta(days=5), 15, 12000],
    [102, datetime.now() - timedelta(days=200), 1, 300],
    [103, datetime.now() - timedelta(days=20), 5, 4500],
    [104, datetime.now() - timedelta(days=2), 2, 18000],
    [105, datetime.now() - timedelta(days=45), 3, 2100]
]
df = pd.DataFrame(data, columns=['ID', 'Last_Purchase', 'Frequency', 'Monetary'])
df.to_csv('loyalty_data.csv', index=False)

df['Recency'] = (datetime.now() - df['Last_Purchase']).dt.days

print("Аналіз клієнтських сегментів:")
print(df[['ID', 'Recency', 'Frequency', 'Monetary']])

plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Recency', y='Frequency', size='Monetary', hue='ID',
palette='viridis', sizes=(100, 1000))
plt.title('Розподіл клієнтів за активністю (RFM)')
plt.xlabel('Днів з останньої покупки (менше = краще)')
plt.ylabel('Кількість замовлень')
plt.grid(True, alpha=0.3)
plt.show()
```

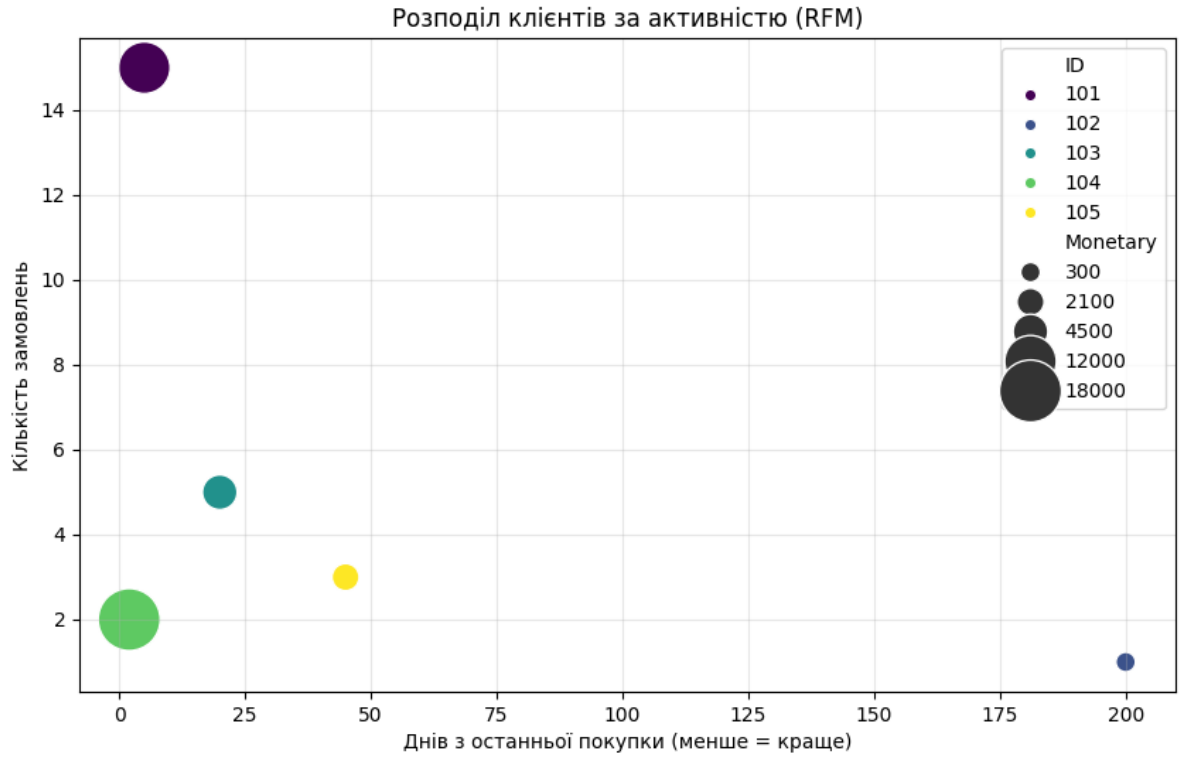


Рисунок 1 – Програмна реалізація аналізу та візуалізації клієнтських сегментів

На представленому графіку кожна точка візуалізує конкретного клієнта, даючи змогу миттєво оцінити стан бази без тривалого вивчення таблиць. Відповідно до отриманих результатів, ми можемо сегментувати аудиторію за їхньою поведінкою. Клієнт 101 виступає «лояльним лідером», маючи найвищу частоту покупок (15) за низької давності (5 днів). Це ядро лояльності, яке забезпечує стабільний дохід. Клієнт 104 є «новою зіркою»: попри лише дві покупки, його фінансовий внесок є найвищим (18 000) за мінімальної давності в 2 дні. Такі користувачі потребують особливої уваги для закріплення їхньої лояльності.

Натомість клієнт 102 з давністю у 200 днів фактично залишив сервіс, а клієнти 103 та 105 перебувають у процесі «остигання» і потребують стимулювання.

Автоматизація такого аналізу безпосередньо в IT-продуктах дає бізнесу можливість діяти на випередження. Замість реакції на падіння прибутку постфактум розробник може налаштувати тригерні сценарії, як-от автоматичне присвоєння VIP-статусів або генерація індивідуальних промокодів для клієнтів, чий показник давності перевищив критичну позначку. Це дає змогу спрямовувати маркетинговий бюджет на найбільш перспективні групи, значно підвищуючи окупність інвестицій.

Проведене дослідження підтверджує, що аналіз даних за допомогою Python є критично важливим для сучасного ритейлу. Поєднання метрик давності, частоти та обсягу витрат дає змогу перетворити сухий лог транзакцій на живий інструмент управління клієнтським досвідом. Це відкриває широкі перспективи для подальшого впровадження систем прогнозування, які зможуть виявляти відтік клієнтів ще до того, як він стане очевидним.

Список використаних джерел

1. From Data to Loyalty: RFM Analysis the Python Way (Medium, 2025). URL: https://medium.com/@nikhilpmarihal9/from-data-to-loyalty-rfm-analysis-the-python-way-b0ac38e6b4f0 (дата звернення: 19.03.2026).

2. The Ultimate Guide To Loyalty Program Analytics (2026 Edition). URL: https://wpswings.com/blog/guide-to-loyalty-program-analytics/ (дата звернення: 19.03.2026).

3. Predictive Customer Churn Analysis in Python: Uncovering Insights for Effective Retention Strategies. URL: https://medium.com/@OlorunwaFagbenro/predictive-customer-churn-analysis-in-python-uncovering-insights-for-effective-retention-57ec8e1d55c4 (дата звернення: 19.03.2026).

4. Алексюк В. В., Січко Т. В. Дослідження особливостей програмного забезпечення для аналізу даних. *Комп'ютерні технології обробки даних:* матеріали Всеукр. наук.-практ. конф. Вінниця, 2022. С. 249–251.

5. Андрійченко К. А., Січко Т. В. Аналіз даних як складова інформаційних технологій. *Комп'ютерні технології обробки даних:* матеріали Всеукр. наук. практ. конф. Вінниця, 2020. С. 8–10.

УДК 004.421:519.6

Клименко Ю. Ю., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

ВЕКТОРИЗАЦІЯ ТЕКСТОВИХ ДАНИХ (TF-IDF) ЯК АЛЬТЕРНАТИВА КОЛАБОРАТИВНІЙ ФІЛЬТРАЦІЇ У РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ та актуальність. Переважна більшість сучасних медіаплатформ будують свої рекомендаційні системи на базі колаборативної фільтрації (Collaborative

Filtering). Суть цього методу полягає в аналізі колективної поведінки та створенні масивних матриць користувацьких взаємодій. Утім на практиці такі рішення часто стикаються з критичним обмеженням, відомим як проблема «холодного старту» (cold start). Якщо сервіс ще не встиг зібрати масштабну базу активних глядачів та історію їхніх оцінок, робоча матриця виходить розрідженою (sparse matrix). Як наслідок, генерувати точні пропозиції стає фактично неможливо. Саме тому сьогодні зростає інтерес до альтернативного підходу – контентної фільтрації (Content-Based Filtering), що спирається на інструменти обробки природної мови (NLP).

Аналіз останніх досліджень. Математичний опис текстів через багатовимірні векторні простори вже давно закріпився як базовий стандарт для інтелектуальних систем. Як свідчать сучасні дослідження, глибокий семантичний аналіз дає змогу знаходити неочевидні паралелі між різними об’єктами. Звичайне ж сортування за стандартними прямолінійними жанровими тегами такі приховані зв’язки просто втрачає.

Мета та постановка завдань. Мета дослідження полягає у проєктуванні рекомендаційного алгоритму для вебдодатка «CineRave». Головна ідея – реалізувати підбір фільмів виключно за рахунок математичного аналізу самого контенту (описів, списку жанрів та слоганів), повністю відмовившись від збору попередньої історії глядацьких оцінок.

Виклад основного матеріалу. Базовим ядром розробленої платформи став алгоритм контентної фільтрації. Щоб підвищити точність фінальних рекомендацій, було вирішено відмовитися від ізольованого аналізу окремих атрибутів. Натомість програмний скрипт зливає сюжетний опис, заявлені жанри та слоган картини в один загальний текстовий масив. Цьому етапу передує обов’язкове очищення даних: регістр символів знижується, а вкладені JSON-подібні структури безпечно розпаковуються вбудованим модулем ast.literal_eval. Далі, щоб перетворити отримані тексти на масив числових векторів, у системі задіяно статистичну міру TF-IDF (Term Frequency – Inverse Document Frequency). Вона працює на основі двох складників. Перший (TF) фіксує частоту конкретного слова в межах одного опису. Другий (IDF) виконує роль своєрідного фільтра: він «штрафує» надто популярні слова (як-от «людина», «життя» чи «світ»), штучно занижуючи їх вагу на фоні всього доступного датасету:

$$IDF(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right),$$

де N – загальний обсяг фільмів у базі.

Завдяки такому підходу специфічні та унікальні терміни набувають найбільшої ваги, перетворюючись на ключові маркери для пошуку. До того ж векторизатор налаштований на роботу з біграмами (параметр ngram_range=(1,2)), що вчить алгоритм розуміти змістовий контекст цілих словосполучень. Пошук найближчих «сусідів» у згенерованому 15 000-вимірному просторі ознак спирається на косинусну подібність. Оскільки результати після TF-IDF-перетворення за замовчуванням вже мають одиничну довжину (L2-нормалізація), стандартну формулу розрахунку вдалося суттєво оптимізувати. Ресурсомістке

ділення на добуток довжин було замінено методом linear_kernel, який обмежується лише швидким скалярним добутком.

Висновки. Застосування статистичної міри TF-IDF разом із математично спрощеним розрахунком косинусної подібності (linear_kernel) виявилося дієвим способом повністю обійти проблему «холодного старту». Створений алгоритм здатний миттєво формувати релевантні добірки кіно завдяки глибокому аналізу семантики. Це робить платформу абсолютно незалежною від розміру глядацької аудиторії та наявності стартової історії лайків чи переглядів.

Список використаних джерел

1. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 3rd ed. New York: Springer, 2022. 1090 p.

2. Rai A., Bhatt P. Content-based Recommendation Engine for Video Streaming Platform. *arXiv*. 2023. URL: https://arxiv.org/abs/2308.08406

3. Scikit-learn: TfidfVectorizer API Reference. 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

4. The Movie Recommendation System using Content Based Filtering with TF-IDF Vectorization and Levenshtein Distance. *IJARSCT*. 2022. URL: https://ijarsct.co.in/Paper3648.pdf

УДК 004.421:519.6

Клименко Ю. Ю., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ОЦІНКА ТА МІНІМІЗАЦІЯ ОБЧИСЛЮВАЛЬНИХ ПОХИБОК В АЛГОРИТМАХ МАШИННОГО НАВЧАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Нарощування масштабу архітектур нейронних мереж експоненційно збільшує обсяг математичних операцій із плаваючою комою. Теоретична елегантність оптимізаційних моделей на практиці неминуче стикається із жорсткими апаратними лімітами щодо представлення дійсних чисел. Як наслідок, генеруються мікроскопічні обчислювальні похибки, що мають тенденцію до накопичення. Їх локалізація та подальша мінімізація постають фундаментальною умовою збереження працездатності алгоритмів, особливо в умовах обмежених серверних потужностей.

Новітні публікації наголошують на деструктивному характері чисельних збоїв. Доведено, що найменші відхилення округлення під час розрахунку активації Softmax здатні ініціювати феномен «Softmax Collapse», який повністю блокує оптимізаційний процес [1]. Відповіддю на ці виклики стала розробка протоколів автоматизованого аудиту машинного коду на предмет чисельної вразливості [2; 5]. Водночас тривають інтенсивні дискусії щодо ефективності форматів змішаної точності [3] та впливу деградації градієнтів на можливості зворотного поширення помилки [4].

Специфіка чинників виникнення машинних неточностей під час ітеративного навчання та формування дієвих стратегій запобігання втраті значущих розрядів враховує такі складники:

1. Механіка виникнення неточностей. Згідно з архітектурою стандарту IEEE 754, будь-яке машинне відображення дійсного числа є строго дискретним. Відсікання та округлення мантиси під час арифметичних дій слугують першопричиною базової обчислювальної нестабільності алгоритмів МН.

2. Акумуляція збурень під час ітерацій. Традиційна процедура градієнтного спуску описується рівнянням:

$$w_{t+1} = w_t - \eta \nabla L(w_t). \quad (1)$$

З огляду на ітеративну специфіку навіть гранично малі відхилення, згенеровані на кожному кроці розрахунку градієнта, сумуються. Згодом це накопичення здатне критично викривити фінальні значення вагових коефіцієнтів.

3. Оцінка продуктивності форматів. Експериментальне зіставлення типів даних фіксує жорсткий компроміс: швидкість виконання операцій завжди масштабується на шкоду стабільності. Узагальнені метрики для архітектур із Tensor Cores наведено в таблиці 1.

Таблиця 1 – Порівняння типів точності в задачах машинного навчання (на основі сучасних GPU з Tensor Cores)

Тип точності	**Швидкість навчання (відносно)**	**Пам'ять (відносно)**	**Стабільність (наявність NaN/Inf)**	**Точність моделі (приклад)**
float32	Базова	Базове	Середня	Базова
float64	Менша на 40–60%	Більше у 2 рази	Висока	Найвища
Mixed Precision (FP16+FP32)	Більша у 1.7–1.9 рази	Менше на 40 %	Залежить від масштабування	Близька до float32
Mixed Precision + Gradient Scaling	Більша у 1.8 раз	Менше на 40 %	Висока	Стабільна

4. Алгоритмічне нівелювання похибок. Тотальний перехід на float64 виключає нестабільність, однак гальмує продуктивність системи. Значно раціональнішим є залучення математичних модифікацій:

– впровадження компенсаційного сумування Кахана під час розрахунку функції втрат для збереження молодших розрядів;

– інтеграція перетворення Log-Sum-Exp для обчислення Softmax:

$$\log \sum_i e^{x_i} = c + \log \sum_i e^{x_i - c}, \quad (2)$$

де $c = \max(x_i)$.

Отже, проблема машинних похибок не вирішується банальним збільшенням апаратної розрядності. Раціональний підбір типів даних, реструктуризація оптимізаційних виразів та інтеграція компенсаційних алгоритмів дають змогу досягти необхідного балансу між стабільністю та енергоефективністю.

Список використаних джерел

1. Grokking at the Edge of Numerical Stability / L. Prieto, M. Barsbey, P. A. M. Mediano, T. Birdal. *arXiv preprint.* arXiv:2501.04697. 2025. 24 p.
2. Automatically Detecting Numerical Instability in Machine Learning Applications via Soft Assertions / S. Sharmin et al. *arXiv preprint.* arXiv:2504.15507. 2025. 16 p.
3. Impact of Mixed Precision Techniques on Training and Inference Efficiency of Deep Neural Networks / R. Zhao et al. *IEEE Access.* 2023. Vol. 11. P. 66056–66068.
4. Karner C., Kazeev V., Petersen P. C. Limitations of neural network training due to numerical instability of backpropagation. *Advances in Computational Mathematics.* 2024. Vol. 50. Article 14.
5. FPCC: Detecting Floating-Point Errors via Chain Conditions / X. Yi et al. *Proceedings of the ACM on Programming Languages (OOPSLA).* 2024.

УДК 004.89

Кобець Д. А., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп'ютерні науки,
Бабаков Р. М., д-р техн. наук, доцент, професор кафедри інформаційних технологій

ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ РИНКОВИХ ДАНИХ ДЛЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ІНВЕСТУВАННЯ ТА ОПТИМІЗАЦІЇ ВИТРАТ

Донецький національний університет імені Василя Стуса, м. Вінниця

В умовах глобальної економічної турбулентності та зростання темпів інфляції для кожного індивіда є критичним питання збереження заощаджень та оптимізація великих витрат. Найважливішим аспектом у цьому є вибір моменту для інвестування або здійснення покупки, що вимагає глибокого аналізу ринкових циклів [1]. Використання технологій Data Science дає змогу створити засоби підтримки прийняття рішень, який допомагає обрати оптимальний час для здійснення транзакцій, мінімізуючи втрати від цінових коливань.

Метою роботи є розробка програмного засобу для аналізу та прогнозування ринкових показників, що дає змогу користувачеві ефективно планувати стратегічні інвестиції та великі покупки з урахуванням макроекономічних факторів та інфляційних ризиків.

Для досягнення мети необхідно реалізувати механізм автоматичного збору даних з відкритих джерел, алгоритми предиктивного аналізу для виявлення ринкових трендів та створити візуальний інтерфейс для демонстрації тенденцій та відображення рекомендацій щодо купівлі.

Пропонована інформаційна система базується на поєднанні методів збору даних (Web Scraping) та їх предиктивного аналізу [2]. Програмний засіб орієнтований на дві ключові функції:

1) Оптимізація стратегічних покупок: система відстежує динаміку цін на дороговартісні товари (наприклад, побутова техніка чи будівельні матеріали),

виявляє сезонні закономірності та попереджає про ймовірні стрибки цін у майбутньому. Це дасть змогу користувачеві вирішувати, чи здійснювати покупку зараз, чи очікувати локального мінімуму.

2) Управління захисними активами: аналіз ринку нерухомості та дорогоцінних металів (золото, платина, срібло) з метою пошуку точок входу для довгострокового збереження заощаджень. Саме ці активи історично демонструють стійкість до знецінення національних валют.

Алгоритмічна база використовує бібліотеки мови Python для аналізу часових рядів. Застосування регресійних моделей дає змогу будувати короткострокові прогнози з урахуванням курсових коливань та ринкових трендів [3]. Користувацький інтерфейс у форматі вебдашборду забезпечує наочну візуалізацію вигоди від прийнятого рішення в порівнянні з поточним рівнем інфляції.

Ґрунтуючись на результатах дослідження, можна стверджувати, що впровадження інтелектуальних систем аналізу ринку в практику особистих фінансів дає змогу звичайному користувачеві застосовувати інструменти професійних інвесторів. Це забезпечує можливість зберігати власний капітал та ефективно протидіяти інфляційним процесам навіть в умовах глобальної економічної нестабільності.

Список використаних джерел

1. Смірнова К. В. Економічний цикл та його сутність. *Вісник студентського наукового товариства ДонНУ імені Василя Стуса*, 2016. Т. 1, № 8. С. 199–203.

2. A practical Introduction to Web Scraping in Python. *Real Python.* URL: https://realpython.com/python-web-scraping-practical-introduction/

3. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. *OTexts.* 2021. URL: https://otexts.com/fpp3/index.html

УДК 004.8

Ліхоткіна В. І., здобувачка вищої освіти 3 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

МЕТОДИ СТАТИСТИЧНОГО АНАЛІЗУ РЕЗУЛЬТАТІВ A/B ТЕСТУВАННЯ У ВЕБСИСТЕМАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

A/B-тестування (розщеплене тестування) – це метод контрольованого експерименту, за якого випадкові групи користувачів одночасно взаємодіють з різними версіями вебресурсу: варіантом A (контроль) та варіантом B (модифікація). Метою є порівняння ефективності цих версій за визначеними метриками для прийняття рішень на основі об'єктивних даних. Статистичний аналіз базується на перевірці нульової гіпотези H_0 (відсутність реальних змін між варіантами) та альтернативної гіпотези H_1 (наявність значущої різниці).

Вибір методу аналізу визначається природою досліджуваної метрики. Для дискретних показників (конверсія, CTR) стандартом є Z-тест. Для неперервних показників (середній чек, час сесії) застосовується t-критерій Велча або непараметричний U-критерій Манна–Уітні, що є стійким до екстремальних відхилень [1]. Ключовими індикаторами прийняття рішень є p-value та довірчі інтервали, які дають змогу оцінити діапазон потенційного ефекту [2]. Під час проведення тестування необхідно враховувати ризики помилок I та II роду, уникаючи «проблеми підглядання» (peaking) [3].

Для практичної перевірки викладених положень наведено приклад реалізації аналізу середнього чека у середовищі R (рис. 1).

```
# 1. Завантаження даних (групи А та В)
set.seed(123)
group_A <- rlnorm(100, meanlog = 6, sdlog = 0.4)
group_B <- rlnorm(100, meanlog = 6.2, sdlog = 0.5)

# 2. Перевірка на нормальність (Тест Шапіро-Вілка)
# Потрібно, щоб зрозуміти, який тест обрати
shapiro.test(group_A)
shapiro.test(group_B)

# 3. Статистичне порівняння (U-критерій Манна-Уітні)
# Обрано цей тест, бо дані мають викиди
test_result <- wilcox.test(group_A, group_B, conf.int = TRUE)
print(test_result)

# 4. Побудова графіка для візуалізації
boxplot(group_A, group_B,
        names = c("Група А", "Група Б"),
        col = c("white", "lightgray"),
        main = "Порівняння результатів A/B тесту",
        ylab = "Сума чека (грн)")
```

Рисунок 1 – Приклад реалізації аналізу середнього чека у середовищі R

Цей приклад демонструє алгоритм вибору критерію залежно від специфіки даних у вебсистемах. Наведений код вирішує типову проблему вебсистем – високий рівень «шуму» та асиметричний розподіл даних. Використання функції rlnorm імітує реальну поведінку користувачів, де одиничні великі замовлення («викиди») викривляють середні значення. Застосування wilcox.test замість стандартних параметричних методів дає змогу отримати достовірний результат p-value незалежно від наявності аномалій.

Графік «ящик з вусами» наочно демонструє зміщення медіани Групи Б та наявність викидів (рис. 2). Поєднання програмного розрахунку та графічного аналізу дає змогу вебсистемам переходити до моделі *Data-Driven,* забезпечуючи точну верифікацію гіпотез у складних інформаційних середовищах.

Ефективність A/B-тестування у вебсистемах залежить не лише від обсягу зібраних даних, а й від коректності вибору методів статистичного аналізу. Використання непараметричного U-критерію Манна–Уітні у поєднанні з мовою програмування R дає змогу нівелювати вплив аномальних викидів, що є характерними для онлайн-трафікf. Цей підхід забезпечує високу точність

верифікації гіпотез, даючи змогу вебсистемам стабільно функціонувати в межах моделі Data-Driven, мінімізувати ризики помилкових рішень та об'єктивно оцінювати вплив будь-яких модифікацій на цільові бізнес-метрики.

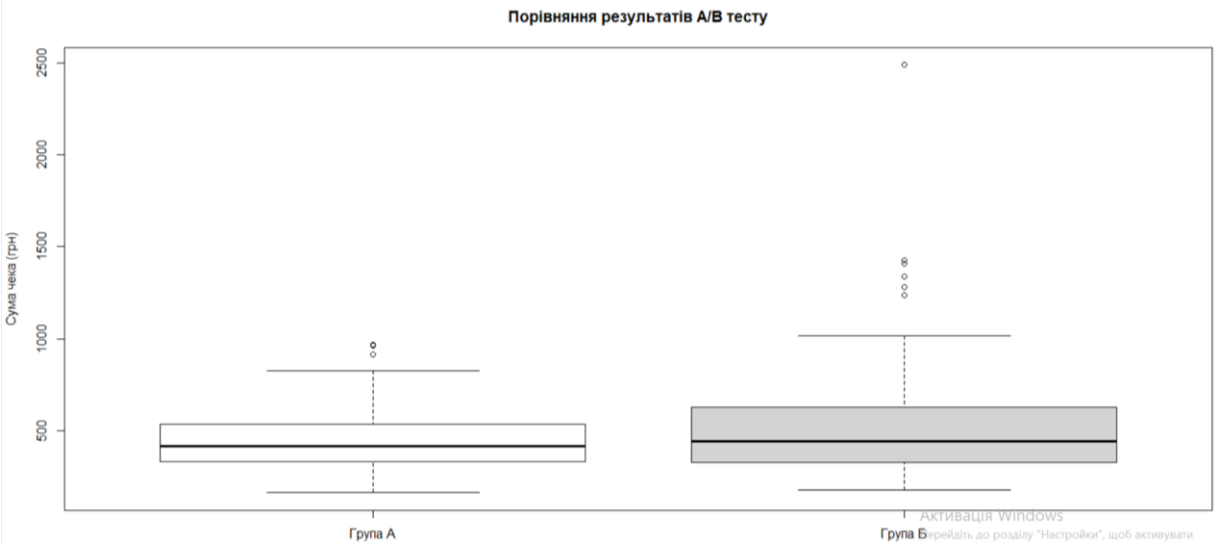


Рисунок 1 – Візуалізація розподілу середнього чека в групах А та Б

Список використаних джерел

1. Kohavi R., Tang D., Xu YTrustworthy Online Controlled Experiments: A Practical Guide to A/B Testing. Cambridge University Press, 2020.
2. Georgiev G. K. Statistical Methods in Online A/B Testing: Statistics for Data-driven Business Decisions and Software Engineering. Analytics Toolkit, 2022.
3. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. OTexts, 2021.

УДК 004.6

Михайляк М. О., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

ВИКОРИСТАННЯ PYTHON ДЛЯ АНАЛІЗУ НЕСТРУКТУРОВАНИХ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі люди щодня створюють величезну кількість інформації. Це повідомлення в соціальних мережах, фотографії, відео, електронні листи, текстові документи та записи з різних сервісів. Більшість таких даних є неструктурованими, тобто не мають чіткої таблиці або впорядкованої форми для зберігання й аналізу. Через це їх складніше обробляти звичайними методами.

Аналіз неструктурованих даних сьогодні став важливим напрямом у сфері інформаційних технологій. Компанії використовують його для пошуку корисної

інформації, прогнозування поведінки користувачів, покращення сервісів та автоматизації процесів. Саме тому зростає потреба у простих і зручних інструментах для роботи з такими даними.

Одним із найпопулярніших інструментів є мова програмування Python [1]. Вона широко використовується у сфері аналізу даних, машинного навчання та штучного інтелекту. Python має простий синтаксис, тому підходить як для початківців, так і для професійних розробників. Ще однією перевагою є велика кількість готових бібліотек, які значно спрощують роботу з інформацією.

Для аналізу текстових даних у Python часто використовують бібліотеки NLTK, spaCy та pandas [2]. Вони дають змогу очищати текст від зайвих символів, знаходити ключові слова, визначати частоту використання слів та аналізувати зміст документів. Наприклад, за допомогою Python можна автоматично визначати тему тексту або знаходити емоційне забарвлення повідомлень у соціальних мережах.

Python також активно застосовується для роботи із зображеннями та відео. Бібліотека OpenCV дає змогу розпізнавати об’єкти, обличчя людей та рух на відео. Такі технології використовують у системах безпеки, медицині, транспорті та промисловості. Наприклад, камери відеоспостереження можуть автоматично виявляти підозрілі дії або сторонні об’єкти.

Важливу роль Python відіграє і в машинному навчанні. Бібліотеки scikit-learn, TensorFlow та Keras допомагають створювати моделі, які здатні аналізувати великі обсяги інформації та знаходити закономірності. Це дає можливість автоматизувати багато процесів і зменшити вплив людського фактора.

Ще однією перевагою Python є можливість працювати з великими обсягами даних у реальному часі. Це особливо важливо для сучасних інформаційних систем, де швидкість обробки інформації має велике значення. Python підтримує інтеграцію з базами даних, вебсервісами та хмарними платформами, що робить його універсальним інструментом для різних сфер діяльності [3].

Отже, Python є ефективним і зручним засобом для аналізу неструктурованих даних. Його можливості дають змогу працювати з текстами, зображеннями, відео та іншими типами інформації. Завдяки простоті використання та великій кількості бібліотек Python залишається однією з основних мов програмування у сфері аналізу даних та штучного інтелекту.

Список використаних джерел

1. Python і аналітика даних. URL: https://itproger.com/ua/news/python-i-data-analytics-chto-nuzhno-znat-i-primer-proekta (дата звернення: 09.05.2026).

2. Бібліотеки для машинного навчання та роботи з даними (Data Science). URL: https://goit.global/ua/articles/top-10-bibliotek-i-freymvorkiv-dlia-ai/ (дата звернення: 09.05.2026).

3. Python у науці про дані. URL: https://buki.com.ua/blogs/python-u-nauci-pro-dani-iak-vikoristovuvati-python-dlia-analizu-danix-ta-masinnogo-navcannia/ (дата звернення: 09.05.2026).

УДК 004.01

Пацула А. О., здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

ІНТЕРАКТИВНИЙ 3D-РЕНДЕРИНГ У ВЕБДИЗАЙНІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ та актуальність. Сучасна індустрія веброзробки переживає трансформацію від статичних двовимірних інтерфейсів до імерсивних тривимірних середовищ. Розвиток технологій WebGL та WebGPU дав змогу перенести потужні графічні обчислення безпосередньо у браузер, відкриваючи нові можливості для презентації продуктів, цифрових двійників (Digital Twins) та інтерактивного навчання. Актуальність теми зумовлена необхідністю пошуку балансу між високою візуальною якістю 3D-сцен та жорсткими обмеженнями вебплатформ щодо швидкості завантаження і споживання апаратних ресурсів користувача.

Аналіз останніх досліджень. Фундаментальні основи рендерингу в браузері закладені стандартом WebGL, який базується на OpenGL ES. Проте складність прямої роботи з низькорівневим API призвела до появи високорівневих бібліотек, як-от Three.js, Babylon.js та PlayCanvas. Дослідження в галузі обчислювальної графіки (зокрема праці Р. Сіммонса та документація Khronos Group) виділяють формат glTF 2.0 як «JPEG для 3D», що став індустріальним стандартом завдяки своїй здатності ефективно передавати геометрію та PBR-матеріали (Physically Based Rendering).

Мета роботи – дослідити архітектуру вебдодатків з інтерактивним 3D-рендерингом, визначити оптимальний пайплайн підготовки hard-surface асетів та проаналізувати методи програмної оптимізації для забезпечення стабільної частоти кадрів (FPS) у браузері.

Викладення основного матеріалу. Робота з 3D у веб кардинально відрізняється від офлайн-візуалізації. Головною проблемою є обмежена пропускна здатність каналів зв'язку та варіативність потужності клієнтських пристроїв. Для вирішення цих завдань пропонується комплексний підхід, що складається з трьох рівнів: геометричного, матеріального та програмного.

1. *Оптимізація геометричних даних.* Для вебдодатків критично важливо використовувати low-poly-моделі. На етапі розробки асета (наприклад, платіжного термінала чи іншого складного пристрою) необхідно проводити ретопологію, щоб мінімізувати кількість трикутників. Вся дрібна деталізація, фаски (bevels) та складні форми мають бути перенесені з high-poly моделі на low-poly через запікання карти нормалей (Normal Map) та карти оклюзії (Ambient Occlusion). Це дає змогу візуально зберегти мільйони полігонів, використовуючи лише кілька тисяч.

2. *Система PBR-матеріалів та пакування текстур.* У вебсередовищі стандарт PBR дає змогу досягти фотореалізму через математичну імітацію поведінки

світла. Для зменшення кількості HTTP-запитів до сервера доцільно застосовувати методику пакування каналів (Channel Packing). Наприклад, карти Roughness, Metalness та AO об'єднуються в один файл (ORM-карта), де кожен параметр займає окремий колірний канал (R, G або B). Використання формату текстур WebP або специфічних розширень glTF (наприклад, KTX2) дає змогу зменшити обсяг відеопам'яті, що споживається.

3. *Архітектура програмного рішення на Three.js.* Побудова сцени базується на графі вузлів (Scene Graph). Основними компонентами є:

– *Renderer:* ініціалізація контексту WebGL та управління циклом анімації (requestAnimationFrame).

– *Camera:* використання PerspectiveCamera для реалістичного сприйняття глибини.

– *Lights:* поєднання статичного оточення (HDRI) та динамічних джерел світла для підсвічування деталей.

– *Raycasting:* метод обчислення перетину променя з об'єктом для обробки кліків користувача по 3D-елементах (наприклад, натискання кнопок на терміналі).

Для підвищення продуктивності в коді треба впроваджувати патерн Instanced Rendering (якщо на сцені багато однотипних об'єктів) та систему рівнів деталізації (LOD – Levels of Detail), яка автоматично замінює модель на спрощену під час віддалення камери. Також доцільно застосовувати Draco-компресію для файлів glB, що дає змогу стискати геометрію до 90 % без візуальних втрат.

Постановка задачі та результати. Під час дослідження було розроблено прототип вебконфігуратора hard-surface-моделі. Практичні тести показали, що застосування Draco-компресії та ORM-пакування текстур зменшило час початкового завантаження сторінки на 65 %. Використання бібліотеки Three.js дало змогу інтегрувати логіку зміни матеріалів та анімації частин об'єкта через об'єктно-орієнтовані класи на TypeScript, що забезпечило чистоту коду та можливість його подальшого розширення.

Для досягнення максимального візуального комфорту користувача та кінематографічної якості зображення застосовуються алгоритми постобробки (Post-Processing). Технічна реалізація цього процесу базується на використанні об'єктів кадрового буфера (Framebuffer Objects, FBO). Замість виводу зображення безпосередньо на екран монітора (canvas) сцена рендериться у віртуальну текстуру, до якої потім послідовно застосовуються додаткові фрагментні шейдери через систему так званих проходів (Render Passes).

Особливе значення для презентації технічних (hard-surface) моделей має інтеграція ефекту SSAO (Screen Space Ambient Occlusion). На відміну від статично запеченої карти AO, алгоритм SSAO динамічно розраховує мікротіні в заглибленнях та на стиках фасок у просторі екрана, що значно підвищує сприйняття об'єму деталей. Оскільки рендеринг геометрії з чіткими краями неминуче супроводжується ефектом «драбинки» (aliasing), обов'язковим є застосування алгоритмів згладжування. Залежно від обчислювальної потужності пристрою користувача конвеєр може використовувати апаратне багатовибіркове згладжування MSAA (Multisample Anti-Aliasing) або продуктивніші шейдерні аналоги, як-от SMAA (Subpixel Morphological Antialiasing) чи FXAA. Додаткові ефекти,

наприклад, емісійне світіння (Bloom) для LED-індикаторів чи екранів терміналів, обробляються окремим проходом з використанням математичного розмиття за Гауссом.

Іншим критично важливим аспектом розробки є архітектура адаптивності (Responsive 3D Design) та управління продуктивністю. Вебдодаток повинен не лише динамічно перераховувати матрицю проєкції та співвідношення сторін камери (aspect ratio) під час зміни розмірів вікна браузера, але й гнучко керувати буфером рендерингу (pixel ratio). Для сучасних дисплеїв з високою щільністю пікселів (наприклад, Apple Retina) рендеринг складних 3D-сцен у нативній роздільній здатності може призвести до критичного перегріву мобільного пристрою та падіння FPS. Тому в архітектуру доцільно закладати алгоритми динамічного масштабування роздільної здатності (Dynamic Resolution Scaling), які автоматично знижують якість рендеру в моменти обчислювальних піків.

Взаємодія з тривимірною сценою (інтерактивність) реалізується через спеціалізовані контролери віртуальної камери (наприклад, OrbitControls). Для забезпечення максимальної плавності рухів (UX) застосовується математичне загасання (damping) на основі сферичної лінійної інтерполяції (Slerp) для кватерніонів. Щоб користувач не зміг вивести об’єкт за межі видимості або «провалитися» всередину геометрії, алгоритм повинен суворо обмежувати сферичні координати камери (azimuth та polar angles), а також дистанцію масштабування. Водночас система має паралельно обробляти як події миші (MouseEvents) для десктопних браузерів, так і мультитач-жести (TouchEvents) для мобільних платформ, гарантуючи безперебійний інтерактивний досвід.

Висновки. Інтерактивний 3D-рендеринг стає невід’ємною частиною сучасного вебдизайну. Найбільш ефективним підходом є використання бібліотеки Three.js у поєднанні з форматом glTF 2.0. Ключем до успіху є жорстка оптимізація асетів на етапі моделювання та запікання текстур, а також програмне управління ресурсами пам’яті через механізми LOD та компресію. Подальший розвиток галузі пов’язаний з переходом на WebGPU, що дасть змогу використовувати ще складніші обчислювальні алгоритми та шейдери безпосередньо у вікні браузера.

Список використаних джерел

1. Dirksen J. Learn Three.js: Program 3D animations and visualizations for the web with HTML5 and WebGL. 4th ed. Birmingham: Packt Publishing, 2023. 620 p.

2. Three.js documentation. URL: https://threejs.org/docs/ (дата звернення: 11.05.2026).

3. glTF 2.0 Specification / Khronos Group. URL: https://registry.khronos.org/glTF/ (дата звернення: 11.05.2026).

4. Кадурін Д. О. Методи оптимізації тривимірних об’єктів для веб-додатків: навч. посіб. Харків: ХНУРЕ, 2022. 112 с.

5. Parisi T. Learning WebGL: Building Real-Time 3D Graphics with WebGL 2.0. 2nd ed. Sebastopol: O’Reilly Media, 2019. 350 p.

УДК 004.891:004.912:330.322

Поліщук В. С., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Ніколюк П. К., д-р фіз.-мат. наук, професор, професор кафедри інформаційних технологій

МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ У СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ ІНВЕСТИЦІЙНИХ РІШЕНЬ ДЛЯ МАЛОГО БІЗНЕСУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасна парадигма цифрової трансформації економіки вимагає переходу від дискретних обчислювальних моделей до інтелектуальних систем, здатних оперувати слабоструктурованою інформацією. Проблема інвестування в об'єкти малого бізнесу характеризується високим рівнем когнітивної невизначеності та інформаційної асиметрії. Традиційний аналітичний інструментарій, обмежений лінійними фінансовими розрахунками, не забезпечує релевантної оцінки якісних детермінант, що зумовлює необхідність впровадження методів інтелектуальної обробки даних у процеси прийняття рішень [1].

Додатковим викликом постає стрімка зміна кон'юнктури ринку та необхідність врахування нефінансових показників, як-от репутаційний капітал, досвід команди та лояльність клієнтського середовища, що часто залишаються поза увагою класичних моделей. Отже, актуальність дослідження зумовлена потребою у розробці інструментарію, що мінімізує суб'єктивізм інвестора та забезпечує верифікацію бізнес-проєктів на основі синергії математичного апарату та інтелектуального семантичного аналізу [2].

У межах дослідження запропоновано гібридний метод оцінювання, інтегрований у програмно-аналітичний комплекс «InvestIQ Pro». Наукова новизна підходу полягає у синергетичному поєднанні класичного економічного моделювання та методів інтелектуального аналізу природної мови (NLP). Технічна архітектура системи реалізована на базі мови Python (мікрофреймворк Flask), що дає змогу забезпечити високу швидкість ітераційної обробки параметрів [3]. Інтелектуальне ядро розробленої системи базується на поєднанні рівня кількісної верифікації, що передбачає реалізацію алгоритмів розрахунку динамічних показників (NPV, PI, DPP) із автоматизованим стрес-тестуванням стійкості моделі, та рівня семантичного скринінгу, який використовує NLP-алгоритми для ідентифікації тональності опису проєкту на основі контент-аналізу семантичних маркерів, що дає змогу конвертувати неструктуровані текстові дані у числовий показник Sentiment Score [4].

Практична верифікація запропонованих методів проводилася шляхом моделювання інвестиційного кейса приватного дитячого садка у м. Києві. Процес підготовки даних та безпосереднього заповнення параметрів у робочій області системи представлено на рис. 1. На цьому етапі інвестор вводить фінансові показники (1,2 млн грн інвестицій та 1,44 млн грн доходу), виставляє рівні ризику повзунками та додає текстовий опис проєкту для NLP-аналізу.



Рисунок 1 – Формування вхідних параметрів об'єкта інвестування в інтерфейсі СППР

На основі введених даних система генерує комплексний аналітичний звіт, візуалізацію якого представлено на рис. 2. За результатами розрахунків встановлено, що проєкт має виняткову привабливість: індекс рентабельності становив PI = 4,02, а термін окупності DPP = 0,8 року. Когнітивна візуалізація у вигляді радарної діаграми та аналіз чутливості NPV підтвердили стійкість бізнесу до ринкових коливань, що разом із високим NLP-балом (100 %) дало змогу системі сформувати вердикт «РЕКОМЕНДОВАНО».

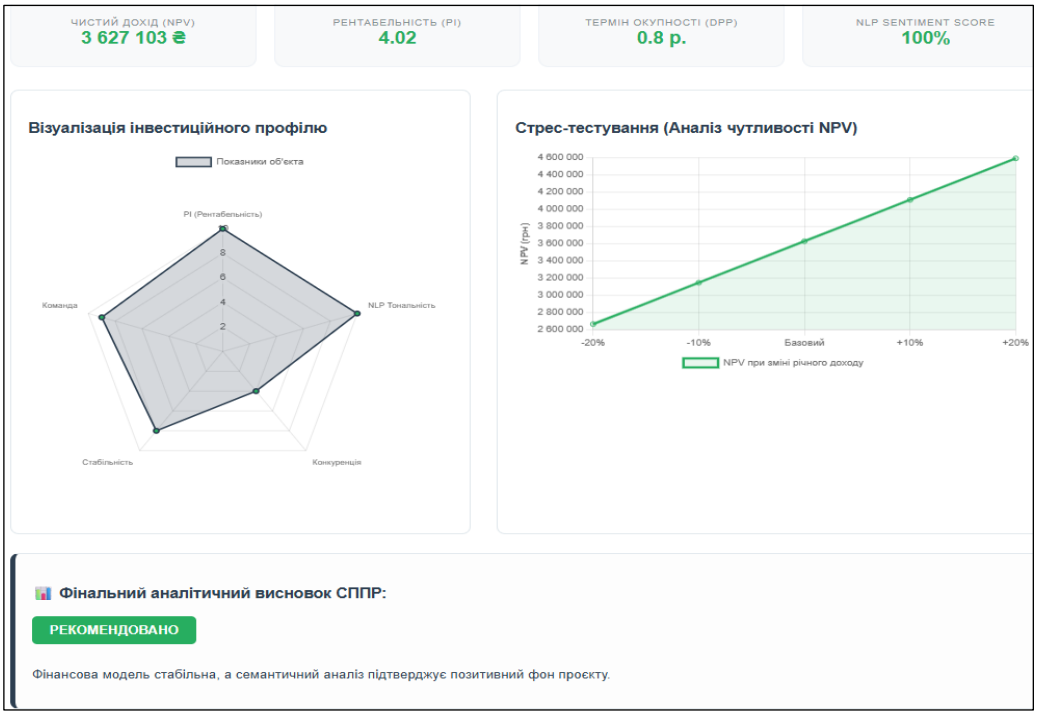


Рисунок 2 – Вихідні результати аналізу та фінальний вердикт системи

Проведене дослідження підтверджує, що інтеграція методів інтелектуальної обробки природної мови та динамічного фінансового моделювання у структуру сучасних систем підтримки прийняття рішень дає змогу докорінно трансформувати процес інвестиційного аналізу. Замість традиційних рутинних розрахунків розроблений підхід забезпечує перехід до високотехнологічної операції багатофакторного оцінювання, де кількісні фінансові метрики (NPV, PI, DPP) гармонійно поєднуються з когнітивним аналізом якісних характеристик бізнес-проєкту. Впровадження алгоритмів семантичного скринінгу (NLP) дає змогу нівелювати критичні когнітивні упередження інвестора та забезпечити високу об'єктивність експертного висновку навіть в умовах значної ринкової невизначеності та інформаційної асиметрії. Побудований програмно-аналітичний комплекс «InvestIQ Pro» не лише автоматизує обчислювальні процеси, а й створює єдиний інтелектуальний простір для верифікації ризиків, що є критично важливим для сегмента малого бізнесу в Україні. Результати практичної апробації продемонстрували, що синергія математичної точності та лінгвістичного аналізу дає змогу ідентифікувати приховані переваги об'єктів інвестування, які зазвичай ігноруються класичним програмним забезпеченням. Отже, запропонований інструментарій є ефективним засобом мінімізації впливу людського фактора, підвищення точності прогнозів та формування стратегічно обґрунтованих рекомендацій, що сприяє розвитку прозорого та цифровізованого інвестиційного середовища.

Список використаних джерел

1. Nouri H., Parker R. J. Decision Support Systems in Small Businesses. ResearchGate, 1991. URL: https://www.researchgate.net/publication/ (дата звернення: 05.05.2026).

2. Yogesha T., Thimmaraju S. N. Leading the Way in Efficient Web Content Mining through Advanced Classification and Clustering Techniques. *International Journal of Intelligent Systems and Applications in Engineering*. 2023. URL: https://ijisae.org/index.php/IJISAE/article/view/2513 (дата звернення: 05.05.2026).

3. Майорова Т. В. Інвестування: навч. посіб. Центр навчальної літератури, 2020. URL: https://buklib.net/books/37224/ (дата звернення: 05.05.2026).

4. Sarker I. H. Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective. SN Computer Science, 2021. URL: https://pubmed.ncbi.nlm.nih.gov/34278328/ (дата звернення: 05.05.2026).

УДК 004.021

Поліщук О. С., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Ніколюк П. К., д-р фіз.-мат. наук, професор, професор кафедри інформаційних технологій

АНАЛІЗ ТА ГЕНЕРАЦІЯ ТЕКСТУ. ПЕРЕКЛАДАЧІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Аналіз і генерація тексту є ключовими напрямами в галузі Text Mining та обробки природної мови (NLP). Основною метою є автоматичне вилучення

змісту з тексту, його інтерпретація та створення нових текстових даних. Одним із найважливіших застосувань є машинний переклад, що дає змогу автоматично перетворювати текст з однієї мови на іншу.

Сучасні системи перекладу базуються на нейронних мережах і здатні забезпечувати якість, близьку до людської для деяких мовних пар [1].

Аналіз тексту включає такі основні етапи:

– токенізація (розбиття на слова);
– лематизація (зведення до початкової форми);
– виділення ознак (TF-IDF, embeddings);
– класифікація та кластеризація.

Одним із показників якості мовної моделі є перплексія (perplexity):

$$Perplexity = 2^{H(p.q)},$$

де $H(p, q)$ – крос-ентропія між реальним і передбаченим розподілом.

Менша перплексія означає кращу якість моделі [2].

Генерація тексту базується на побудові мовних моделей, які прогнозують наступне слово. Сучасні системи використовують архітектуру Transformer, яка базується на механізмі self-attention [3].

Формально задача генерації:

$$P(y_1, y_2, \dots, y_n) = \prod_{t=1}^{n} P(y_t | y_1, \dots, y_t - 1).$$

Аналітично це означає, що кожне слово залежить від попереднього контексту.

Сучасні моделі, як-от ChatGPT, здатні:

– генерувати тексти різних стилів;
– відповідати на запитання;
– виконувати переклад і узагальнення.

Машинний переклад пройшов кілька етапів розвитку [3]:

1) правило-орієнтований переклад;
2) статистичний переклад;
3) нейронний переклад (NMT).

Сучасні системи використовують саме нейронний підхід, який базується на ймовірнісній моделі [2]:

$$P(y \mid x),$$

де x – текст мовою-джерелом;

y – переклад.

Аналітичні переваги NMT:

– врахування повного контексту речення;
– краща обробка граматики;
– зменшення помилок узгодження.

Однак існують і недоліки:

– складність інтерпретації моделей;

– потреба у великих обсягах даних;
– помилки під час перекладу рідкісних мов.

BLEU (Bilingual Evaluation Understudy) [4]:

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} \omega_n \log p_n\right).$$

Ця метрика оцінює схожість між машинним і еталонним перекладом.

Аналітичні обмеження BLEU:

– не враховує синонімію;
– не оцінює зміст;
– залежить від еталонного перекладу.

Тому сьогодні використовуються додаткові метрики:

- ROUGE;
- METEOR;
– BERTScore.

Сучасний розвиток пов’язаний із використанням великих мовних моделей (LLM) [4; 5]:

- інтеграція перекладу в універсальні моделі;
- мультимодальність (текст + зображення);
- адаптація до стилю користувача.

Системи, як-от ChatGPT, поєднують функції перекладу, аналізу та генерації тексту [3].

На рис. 1 також видно відмінність між двома ключовими підходами до обробки тексту – моделями типу BERT та GPT, які базуються на архітектурі Transformer architecture.

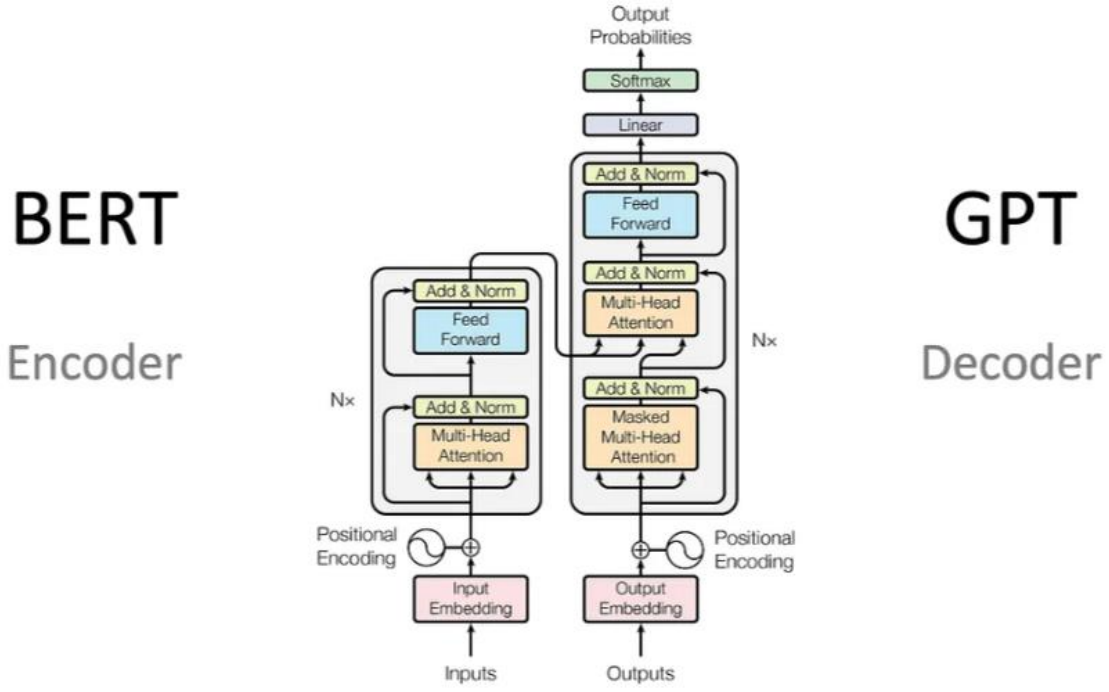


Рисунок 1 – Архітектура Transformer (енкодер-декодер)

BERT використовує енкодер і працює двонаправлено, аналізуючи контекст з обох сторін, тому ефективний для задач аналізу тексту [1]. GPT використовує декодер і генерує текст послідовно, що робить його придатним для створення тексту та перекладу.

Механізм Multi-Head Attention дає змогу враховувати контекст речення, а повторення блоків (Nx) покращує якість обробки тексту [3].

Отже, Transformer-архітектура є основою сучасних систем аналізу та генерації тексту. Загалом сучасний розвиток аналізу та генерації тексту базується на застосуванні нейронних мереж та архітектури Transformer, що забезпечує високий рівень точності обробки природної мови та машинного перекладу. Використання механізмів self-attention і багатоголової уваги дає змогу системам враховувати контекст у реченні. Моделі типу BERT показують високу продуктивність у задачах аналізу тексту, оскільки двонаправлений енкодер допомагає враховувати контекст з обох сторін.

Список використаних джерел

1. Jurafsky D., Martin J. H. Speech and Language Processing. 3rd ed. 2023. URL: https://web.stanford.edu/~jurafsky/slp3/

2. Neural Machine Translation. *Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/wiki/Neural_machine_translation

3. Attention Is All You Need / A. Vaswani, N. Shazeer, N. Parmar et al. 2017. URL: https://arxiv.org/abs/1706.03762

4. Koehn P. Neural Machine Translation. Cambridge University Press, 2020.

5. BLEU. *Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/wiki/BLEU

УДК 004.422.5.

Ратушний О. С., здобувач вищої освіти 3 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПОШУКУ ОПТИМАЛЬНОГО ШЛЯХУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Задача пошуку оптимального шляху в графі є однією з ключових у комп'ютерних науках. Вона застосовується в навігаційних системах, робототехніці, логістиці та розробці відеоігор. Для її розв'язання існує ціла низка алгоритмів – Дейкстри, A*, BFS, Bellman-Ford та інші, – кожен з яких має свої переваги та обмеження залежно від типу графу і умов задачі [1]. На практиці вибір алгоритму покладається на розробника, що вимагає глибоких знань і значних часових затрат. Застосування методів машинного навчання дає змогу автоматизувати цей вибір, підвищуючи як ефективність розробки, так і якість отриманого рішення.

Кожен алгоритм пошуку шляху орієнтований на певні умови. Алгоритм Дейкстри гарантує оптимальний результат у зважених графах без від'ємних

ребер, але є повільним на великих графах. A* пришвидшує пошук завдяки евристичній функції, проте його ефективність критично залежить від якості обраної евристики [1]. BFS оптимальний для незважених графів, тоді як Bellman-Ford необхідний за наявності від'ємних ваг [2].

На практиці розробник змушений вручну аналізувати характеристики графу – його розмір, щільність, наявність і знак ваг, динамічність середовища – і на основі цього обирати алгоритм та налаштовувати його параметри. Такий підхід є трудомістким, вимагає експертних знань і залишає місце для суб'єктивних помилок. Особливо гостро ця проблема проявляється в системах, де тип задачі змінюється динамічно і щоразу потребує іншого рішення [2].

Задачу вибору алгоритму пошуку шляху можна сформулювати як задачу класифікації: на вході – характеристики графу та умови задачі, на виході – рекомендований алгоритм. Для цього модель машинного навчання навчається на наборі прикладів, де для кожної конфігурації графу зафіксовано, який алгоритм показав найкращий результат за часом виконання та якістю шляху [3].

Такий підхід має кілька переваг перед ручним вибором. По-перше, навчена модель приймає рішення миттєво. По-друге, вона здатна враховувати складні залежності між параметрами графу, які важко формалізувати вручну. По-третє, модель можна донавчати на нових даних, що робить систему гнучкою та масштабованою [3; 4]. Зокрема, методи на основі дерев рішень і градієнтного бустингу показують високу точність у подібних задачах класифікації алгоритмів [4].

Автоматизований вибір алгоритму пошуку шляху є актуальним у багатьох галузях. У розробці відеоігор різні сцени можуть мати принципово різну структуру графу, тому інтелектуальна система здатна динамічно обирати найефективніший алгоритм для кожної з них [1]. У логістиці та навігаційних системах умови задачі постійно змінюються: розмір мережі, наявність обмежень, пріоритет швидкості чи оптимальності – і МН-модель може адаптуватись до цих змін без участі людини [4]. У робототехніці такий підхід дає змогу автономному агенту самостійно обирати стратегію навігації залежно від поточного середовища.

Отже, ручний вибір алгоритму пошуку оптимального шляху є суттєвою практичною проблемою, яка потребує глибоких знань і значних часових витрат, а в умовах динамічно змінюваних задач стає ще більш складною. Застосування методів машинного навчання дає змогу автоматизувати цей процес, формулюючи його як задачу класифікації за характеристиками графу. Такий підхід скорочує час прийняття рішення, мінімізує вплив людського фактора і відкриває можливості для побудови адаптивних інтелектуальних систем пошуку шляху в різних прикладних галузях.

Список використаних джерел

1. A Systematic Review and Analysis of Intelligence-Based Pathfinding Algorithms in the Field of Video Games / T. Standley та ін. *Applied Sciences.* 2022. Vol. 12, № 11. P. 5499.

2. De Sirisuriya S. C. M. S., Fernando T. G. I., Ariyaratne M. K. A. Algorithms for path optimizations: a short survey. *Computing.* 2023. Vol. 105. P. 293–319.

3. MAPFASTER: A Faster and Simpler Take on Multi-Agent Path Finding Algorithm Selection / J. Alkazzi, A. Rizk, M. Salomon, A. Makhoul. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022).* Kyoto: IEEE, 2022. P. 10088–10093.

4. A multi-algorithm pathfinding method: exploiting performance variations for enhanced efficiency / M. Parooei, M. T. Masouleh, A. Kalhor та ін. *Annals of Mathematics and Artificial Intelligence.* 2024. URL: https://doi.org/10.1007/s10472-024-09957-3 (дата звернення: 10.05.2025).

5. Штучний інтелект у науці та освіті: матеріали Міжнар. наук. конф. (AISE 2024), м. Київ, 1–2 бер. 2024 р. Київ, 2024. Секція «Нейронні мережі та машинне навчання».

УДК 681.5.015:007

Русавський О. О., аспірант 2 року навчання спеціальності 122 Комп'ютерні науки,
Роштейн О. П., д-р техн. наук., професор, професор кафедри інформаційних технологій

ЗАСТОСУВАННЯ НЕЧІТКИХ КОГНІТИВНИХ КАРТ ДЛЯ ПРОГНОЗУВАННЯ ТА ДІАГНОСТИКИ В УЗАГАЛЬНЕНИХ ДИНАМІЧНИХ СИСТЕМАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Вступ. Узагальнена динамічна система (УДС), за визначенням В. М. Глушкова [1], – це сукупність взаємопов'язаних об'єктів і процесів, які змінюються в часі. До класу УДС належать промислові підприємства, живі організми, бойові дії та інші системи в економіці, соціології, екології тощо, які характеризуються великою кількістю взаємодіючих змінних і невизначеністю інформації, що часто має експертний характер. У роботі [2] показано, що нечіткі когнітивні карти (НКК) Б. Коско [3] є зручним засобом розробки моделей для сценарного аналізу УДС на основі мінімальної інформації.

Прийняття рішень в УДС пов'язане із задачами прямого і зворотного висновку. Прямий висновок – це погляд у майбутнє, що полягає у прогнозуванні наслідків (виходів системи) за відомими причинами (входами). Зворотний висновок – це погляд у минуле, тобто діагностика причин, які призводять до наслідків, доступних для спостереження в експерименті.

Метою роботи є розробка моделей та алгоритмів для вирішення задач прогнозування та діагностики в УДС шляхом інтеграції НКК та генетичного алгоритму. НКК розглядається як модель «багато входів – багато виходів» для прогнозування станів системи, а генетичний алгоритм застосовується для мінімізації відстані між прогнозними і експериментальними векторами наслідків.

1. ПРОГНОЗУВАННЯ НА ОСНОВІ НКК

НКК являє собою орієнтований граф зі зваженими дугами. Вершини графу, що називаються концептами, відповідають змінним, які враховуються в моделі, а ваги дуг відображають сили впливів змінних-причин на змінні-наслідки.

Для моделювання динаміки зміни значень концептів використовуються такі поняття: матриця сил впливів $\boldsymbol{W}_0$, початковий стан $\boldsymbol{X}_0$, стаціонарний стан $\boldsymbol{X}_l$ та відома множина концептів $C = \{C_1, C_2, \dots, C_n\}$. Значення концептів вираховуються за допомогою рекурентного співвідношення в матричній формі [4]:

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k \oplus (\boldsymbol{X}^k \ominus \boldsymbol{X}^{k-1})\boldsymbol{W}_0. \quad (1)$$

Алгоритм прогнозування:

Крок 1. Задати вектор значень вхідних змінних

$$\boldsymbol{X}_U = [x_1, x_2, \dots, x_q]. \quad (2)$$

Крок 2. Задати початковий стан НКК (2) вектором

$$\boldsymbol{X}^0 = [x_1^0 = x_1, x_2^0 = x_2, \dots, x_q^0 = x_{q,}, x_{q+1}^0 = x_{q+2}^0, \dots, = x_n^0 = -1]. \quad (3)$$

Крок 3. Користуючись рекурентним співвідношенням (1) та початковим вектором (3), обчислити вектор значень концептів в стаціонарному стані

$$\boldsymbol{X}^l = [x_1^l, x_2^l, \dots, x_q^l, x_{q+1}^l, x_{q+2}^l, \dots, x_n^l]. \quad (4)$$

Крок 4. Нормалізувати елементи вектора (4) та отримати вектор

$$\widehat{\boldsymbol{X}} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_q, \hat{x}_{q+1}, \hat{x}_{q+2}, \dots, \hat{x}_n], \quad (5)$$

де $\hat{x}_i = \begin{cases} \frac{x_i^l}{\overline{x}_i}, \text{якщо } x_i^l > 0 \\ -\frac{x_i^l}{\underline{x}_i}, \text{якщо } x_i^l < 0 \end{cases}$;

$\underline{x}_i(\overline{x}_i)$ – мінімальне (максимальне) значення концепту C_i в стаціонарному стані.

Крок 5. Із елементів вектора (5) отримати вектор

$$\widehat{\boldsymbol{X}}_G = [\hat{x}_{q+1}, \hat{x}_{q+2}, \dots, \hat{x}_n], \quad (6)$$

який є прогнозом значень вихідних змінних, що відповідають вхідному вектору (2).

2. ДІАГНОСТИКА НА ОСНОВІ НКК

Зворотний висновок розглядається як задача оптимізації в межах моделі «багато входів – багато виходів», де виходи – це наслідки, що спостерігаються в експерименті, а входи – це причини, які необхідно розпізнати.

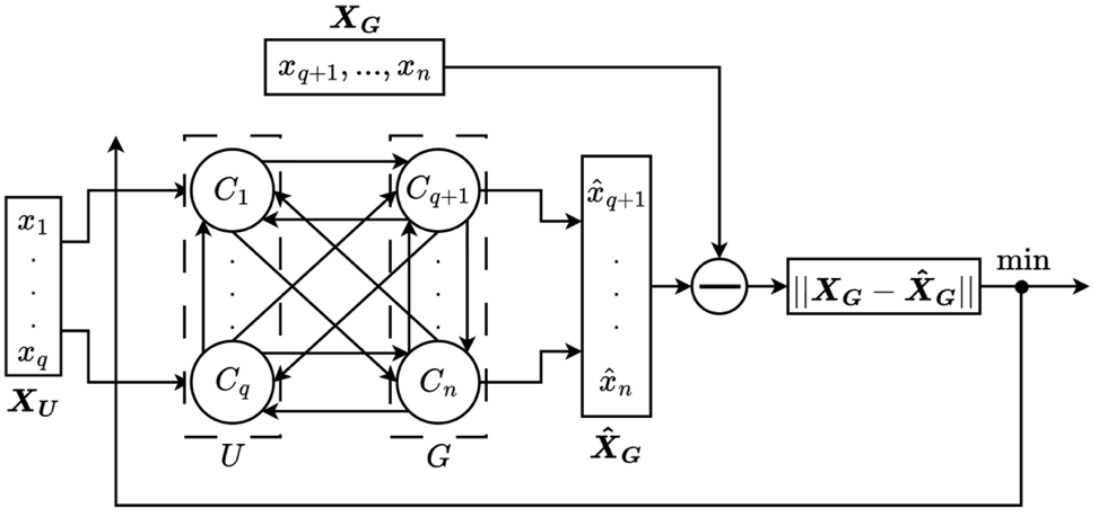


Рисунок 1 – Блок-схема зворотного висновку

Для відновлення причин по відомим наслідкам необхідно знайти такий вектор $\boldsymbol{X}_U$, який забезпечує мінімальну розбіжність між експериментальним ($\boldsymbol{X}_G$) та прогнозованим ($\widehat{\boldsymbol{X}}_G$) векторами:

$$\left\|\boldsymbol{X}_G - \widehat{\boldsymbol{X}}_G\right\| \underset{\boldsymbol{X}_U}{\rightarrow} \min. \quad (7)$$

Для вирішення задачі оптимізації використовується генетичний алгоритм [4; 5].

3. ПРИКЛАД ЗАСТОСУВАННЯ

Типовим представником УДС є автопідприємство. Для моделювання надійності визначено вхідні концепти та вихідні концепти [2].

Результати прогнозування (розділ 1) та діагностики (розділ 2) підприємства для 4 сценаріїв (S) наведені в табл. 1, де:

$[x_1, \dots, x_5]$ – еталонні значення концептів-причин, які відповідають вхідному вектору $\boldsymbol{X}_U$ (2);

$[\hat{x}_6, \dots, \hat{x}_{10}]$ – прогнозовані значення концептів-наслідків, що відповідають вихідному вектору $\widehat{\boldsymbol{X}}_G$ (6);

$[x_1^*, \dots, x_5^*]$ – отримані значення концептів-причин.

Таблиця 1 – Результати прогнозування та діагностики

S	**Еталонні значення концептів-причин** $[\boldsymbol{x_1, \dots, x_5}]$	**Значення концептів-наслідків** $[\boldsymbol{\hat{x}_6, \dots, \hat{x}_{10}}]$	**Отримані значення концептів-причин** $[\boldsymbol{x_1^*, \dots, x_5^*}]$
s_1	0.0,-0.6,0.0,-0.8,-0.6	-0.206,-0.378,0.341,-0.492,-0.438	-0.088,-0.664,-0.043,-0.826,-0.508
s_2	0.4,-0.2,-0.4,-0.4,0.2	0.794,0.719,0.664,1.000,0.890	0.346,-0.183,-0.416,-0.524,0.495
s_3	0.8,0.4,-0.8,0.2,0.6	0.695,0.726,0.326,1.000,0.890	0.764,0.525,-0.863,0.213,0.574
s_4	0.8,0.8,-0.8,0.6,0.8	0.664,0.735,0.226,1.000,0.890	0.606,0.826,-0.827,0.562,0.799

Кількісні оцінки подібності результатів зворотного висновку та еталонних значень концептів-причин за критеріями MSE (mean square error) та MAD (mean absolute deviation) наведені в табл. 2, де:

$$MSE = \frac{1}{5}[(x_1 - x_1^*)^2 + \dots + (x_5 - x_5^*)^2],$$

$$MAD = \frac{1}{5}[|x_1 - x_1^*| + \dots + |x_5 - x_5^*|].$$

Таблиця 2 – Оцінка якості результатів зворотного висновку

S	**MSE**	**MAD**
s_1	0.005	0.063
s_2	0.021	0.101
s_3	0.004	0.052
s_4	0.008	0.057

Висновки. Запропоновано підхід до прогнозування та діагностики УДС шляхом інтеграції НКК та генетичного алгоритму. Для задачі прогнозування НКК забезпечує розрахунок прогнозних значень усіх взаємопов'язаних концептів системи. Діагностика зводиться до задачі оптимізації, яка вирішується за допомогою генетичного алгоритму. Ефективність методів прогнозування та діагностики проілюстрована на прикладі автомобільного підприємства

Список використаних джерел

1. Глушков В. М. Введение в АСУ. Киев: Техника, 1974. 340 с.
2. Нечіткий сценарний аналіз узагальнених динамічних систем / О. П. Ротштейн, А. А. Кашканов, О. В. Зелінська, Д. І. Катєльніков. *Кібернетика та системний аналіз.* 2026, Т. 62, № 1. С. 89–99. DOI: https://doi.org/10.34229/KCA2522-9664.26.1.8
3. Kosko B. Fuzzy cognitive maps. *International Journal of Man-Machine Studies.* 1986, Vol. 24. P. 65–75. DOI: https://doi.org/10.1016/S0020-7373(86)80040-2
4. Rotshtein A., Katielnikov D. Fuzzy Cognitive Map vs Regression. *Cybernetics and Systems Analysis.* 2021. Vol. 57, № 1. P. 605–616. DOI: https://doi.org/10.1007/s10559-021-00385-3
5. Fuzzy cognitive map and mean square method in empirical modelling: application in economics / A. Rotshtein, A. Yosef, T. Neskorodeva, D. Katielnikov. *Expert systems with applications.* 2024. Vol. 247. P. 123176. https://doi.org/10.1016/j.eswa.2024.123176

УДК 004.422.5.

Слободяник Р. Р., здобувач вищої освіти 3 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

МЕТОДИ МАШИННОГО НАВЧАННЯ У ПРОГНОЗУВАННІ ЧАСОВИХ РЯДІВ: ТРАНСФОРМЕРИ ПРОТИ КЛАСИЧНИХ ПІДХОДІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Прогнозування часових рядів є однією з ключових задач машинного навчання, що знаходить застосування у фінансовій аналітиці, метеорології, енергетиці та охороні здоров'я. Протягом десятиліть домінуючою парадигмою залишалися статистичні методи на кшталт ARIMA, однак з розвитком глибокого навчання їм на зміну прийшли нейронні архітектури – спершу рекурентні мережі, а згодом трансформери, що адаптували механізм уваги з обробки природної мови до задач прогнозування [1]. Попри широкий ентузіазм наукової спільноти, низка досліджень засвідчила, що складні трансформер-моделі не завжди перевершують класичні підходи, що робить порівняльний аналіз цих методів актуальним науковим завданням.

ARIMA (AutoRegressive Integrated Moving Average) є одним із найбільш усталених статистичних методів прогнозування, що моделює лінійні залежності між спостереженнями ряду. Метод характеризується прозорою математичною формалізацією, низькими обчислювальними витратами та задовільною точністю

на коротких горизонтах прогнозування [2]. Водночас він не здатний фіксувати нелінійні залежності та складні мультиваріатні взаємозв'язки, що обмежує його застосовність у сучасних задачах великого масштабу.

N-BEATS (Neural Basis Expansion Analysis for Interpretable Time Series Forecasting) є нейромережевим методом, що долає зазначені обмеження завдяки ієрархічній архітектурі на основі fully-connected-блоків із механізмом залишкових зв'язків [3]. N-BEATS демонструє стабільно високу точність на численних бенчмаркових наборах даних, не потребуючи зовнішніх ознак та складної попередньої обробки даних, що робить його практично зручним інструментом.

Informer був запропонований як ефективна адаптація трансформера для довгострокового прогнозування з використанням механізму ProbSparse self-attention, що знижує обчислювальну складність з $O(L^2)$ до $O(L \log L)$ [4]. Autoformer розвиває цю ідею, замінюючи стандартний механізм уваги на Auto-Correlation, який виявляє сезонні залежності через автокореляційний аналіз у частотній області [1]. Обидві архітектури продемонстрували суттєві переваги над рекурентними мережами на довгих горизонтах прогнозування.

Незважаючи на архітектурну складність, дослідження Zeng et al. (2023) показало, що прості лінійні моделі здатні перевершити Informer і Autoformer на стандартних бенчмарках довгострокового прогнозування [5]. Причиною є відсутність у трансформерів вбудованого індуктивного зміщення для послідовних залежностей та схильність до перенавчання на часових рядах із невисоким рівнем складності.

Вибір між трансформерними і класичними методами прогнозування визначається кількома ключовими факторами: горизонтом прогнозування, обсягом навчальних даних та рівнем нелінійності ряду. На коротких горизонтах і з обмеженими даними ARIMA залишається конкурентоспроможним, а N-BEATS – оптимальним нейромережевим вибором [3]. Трансформери доцільно застосовувати за великих обсягів мультиваріатних даних з нелінійними міжзмінними залежностями, де механізм уваги реалізує свій потенціал повною мірою.

Актуальним напрямом розвитку є гібридні архітектури, зокрема PatchTST, що застосовує механізм уваги до патчів часового ряду замість окремих кроків, досягаючи балансу між виразністю та обчислювальною ефективністю [2]. Такий підхід дає змогу суттєво скоротити довжину вхідної послідовності для механізму уваги і підвищити узагальнювальну здатність моделі.

Отже, прогнозування часових рядів методами машинного навчання охоплює широкий спектр підходів – від класичних статистичних до сучасних трансформер-архітектур. Результати порівняльних досліджень свідчать, що трансформери не є безумовно кращими: на низці задач прості моделі демонструють вищу або еквівалентну точність за значно менших обчислювальних витрат. Вибір методу прогнозування повинен ґрунтуватися на характеристиках конкретної задачі – горизонті, обсязі та структурі даних – а не лише на архітектурній складності моделі.

Список використаних джерел

1. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting / H. Wu, J. Xu, J. Wang, M. Long. *Advances in Neural Information Processing Systems.* 2021. Vol. 34. P. 22419–22430.

2. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers / Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam. *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023).* Kigali: OpenReview, 2023.

3. N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting / B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio. *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020).* Addis Ababa: OpenReview, 2020.

4. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting / H. Zhou та ін. *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021).* Vancouver: AAAI Press, 2021. P. 11106–11115.

5. Are Transformers Effective for Time Series Forecasting? / A. Zeng, M. Chen, L. Zhang, Q. Xu. *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI 2023).* Washington: AAAI Press, 2023. P. 11121–11128.

УДК 004.855.5:336.71

Тищенко О. М., здобувач вищої освіти 3 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

ІНТЕЛЕКТУАЛЬНА СИСТЕМА АНАЛІЗУ БАНКІВСЬКИХ ВИПИСОК З ВИЯВЛЕННЯМ АНОМАЛЬНИХ ВИТРАТ ТА ПРОГНОЗУВАННЯМ ФІНАНСОВИХ ПОКАЗНИКІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасних умовах цифровізації фінансового сектору автоматизований аналіз банківських транзакцій набуває особливої актуальності. Зростання обсягів платіжних даних унеможливлює їх ручну обробку, тоді як виявлення аномальних операцій і прогнозування витрат дають змогу своєчасно реагувати на фінансові ризики. Поєднання статистичних і машинних методів у єдиному програмному засобі забезпечує гнучкість та достовірність аналізу.

Інтелектуальна система аналізу банківських виписок реалізує функціонал автоматичного очищення транзакційних даних, виявлення аномалій двома незалежними методами та прогнозування витрат на короткостроковий і середньостроковий горизонти. Система імпортує CSV-файл із колонками Date, Category, Amount. Після відбору витратних транзакцій (від'ємні значення Amount) формується набір ознак: абсолютна сума, логарифм суми, день тижня, тиждень, місяць, квартал, синусно-косинусні компоненти сезонності та відносна величина витрати в категорії.

Перший метод – робастна Z-оцінка. Для кожної транзакції обчислюється глобальна оцінка z_global на основі медіани та медіанного абсолютного відхилення (MAD) логарифмованих сум, а також категоріальна оцінка z_cat. Транзакція визнається аномальною за умови:

```
|z_global| ≥ 2.5 або relative_amount ≥ 2.5 або (|z_cat| ≥ 2.5 і n_cat ≥ 4)
```

Другий метод – Isolation Forest (sklearn, n_estimators = 400). Алгоритм ізолює рідкісні спостереження у просторі ознак без попередньої розмітки. Поріг аномальності встановлюється на рівні 3-го перцентиля scores. Комбінований підхід підвищує пояснюваність: транзакцію позначають аномальною, якщо обидва методи дають позитивний результат, або якщо один із них фіксує яскраво виражений відхил.

Прогнозування витрат ґрунтується на щоденній агрегації витрат і моделі RandomForestRegressor. Модель навчається на часових ознаках і виконує прогноз на 7 днів, 30 днів та 6 місяців. Перевага ансамблевого регресора – здатність враховувати нелінійні сезонні залежності без явного задання їх форми.

Систему протестовано на наборі з 50 синтетичних транзакцій, що містять штучно внесені аномалії різних типів: одиничний великий платіж, аномальна серія в одній категорії, нетипова денна активність. Результати наведено в табл. 1.

Таблиця 1 – Результати тестового аналізу банківських транзакцій

Метод	**Поріг виявлення**	**Виявлено аномалій**	**Збіг між методами**
Робастна Z-оцінка	$\|z\| \geq 2.5$ або відносна сума ≥ 2.5	7 з 50	100 %
Isolation Forest	3-й перцентиль score (n = 400)	7 з 50	100 %

Джерело: сформовано авторами

Обидва методи виявили однакову кількість аномалій із повним збігом, що підтверджує узгодженість підходів і стабільність системи. Графічний інтерфейс (Tkinter) забезпечує вибір файла, запуск аналізу одним кліком та збереження звіту у форматі PDF.

Висновки. Розроблена система поєднує попередню обробку даних, робастну статистику та машинне навчання в єдиному конвеєрі. Комбінування Z-оцінки та Isolation Forest підвищує точність і пояснюваність результатів виявлення аномалій. Прогнозування на основі RandomForestRegressor дає змогу отримати коректний короткостроковий та середньостроковий прогноз витрат. Система є практично придатною для персонального фінансового моніторингу та малого бізнесу.

Список використаних джерел

1. Scikit-learn: Machine Learning in Python / F. Pedregosa et al. *Journal of Machine Learning Research.* 2011. Vol. 12. P. 2825–2830.

2. Liu F. T., Ting K. M., Zhou Z.-H. Isolation Forest. Proc. *8th IEEE Int. Conf. on Data Mining.* 2008. P. 413–422.

3. Breiman L. Random Forests. Machine Learning. 2001. Vol. 45. P. 5–32.

4. Rousseeuw P. J., Croux C. Alternatives to the Median Absolute Deviation. *Journal of the American Statistical Association.* 1993. Vol. 88. № 424. P. 1273–1283.

УДК 004.8:656.07

Уманська А. В., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Ніколюк П. К., д-р техн. наук., професор, професор кафедри інформаційних технологій

МАТЕМАТИЧНІ МОДЕЛІ, АЛГОРИТМИ ТА МЕТОДИ ШІ В ТРАНСПОРТНІЙ ЛОГІСТИЦІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Транспортна логістика як складник управління ланцюгами постачання базується на формалізації процесів перевезення у вигляді математичних моделей та алгоритмів. Основними задачами є задача маршрутизації транспортних засобів (Vehicle Routing Problem), задача комівояжера (Traveling Salesman Problem), задача управління запасами (Inventory Management) та задача розподілу ресурсів.

Класичною моделлю управління запасами є модель економічного розміру замовлення (Economic Order Quantity), відома як модель Уїлсона:

$$Q = \sqrt{\frac{2KD}{h}},$$

де Q – оптимальний розмір замовлення;

K – витрати на оформлення замовлення;

D – попит;

h – витрати на зберігання одиниці продукції.

Для задач маршрутизації використовується модель VRP, яка формалізується як задача мінімізації функції вартості:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij},$$

де c_{ij} – вартість переміщення між вузлами i та j;

$x_{ij} \epsilon \{0,1\}$ – змінна, що визначає факт переміщення.

Задача VRP є NP-складною, тому для її розв'язання застосовуються як точні, так і наближені методи. До точних методів належать:

- метод гілок і меж;
- метод гілок і цін;
- цілочисельне лінійне програмування.

Серед евристичних і метаевристичних методів найбільш поширені:

- алгоритм найближчого сусіда;
- алгоритм Кларка–Райта;
- генетичні алгоритми;
- алгоритм імітації відпалу;

– табу-пошук;

– алгоритм мурашиної колонії.

Для задач найкоротшого шляху використовуються класичні графові алгоритми:

– алгоритм Дейкстри;

– алгоритм Беллмана–Форда;

– алгоритм A* (A-star), який широко застосовується в динамічній маршрутизації.

Час транспортування визначається базовою залежністю:

$$t = \frac{L}{V},$$

де L – довжина маршруту;

V – швидкість руху.

Собівартість перевезення визначається лінійною моделлю:

$$C = C_f + c \cdot L,$$

де C_f – фіксовані витрати;

c – змінні витрати на одиницю відстані.

У сучасних логістичних системах значного розвитку набули методи штучного інтелекту, зокрема машинного навчання. Вони використовуються для задач прогнозування попиту, оцінки часу доставки та оптимізації маршрутів у реальному часі.

До основних методів машинного навчання, що застосовуються в логістиці, належать:

– лінійна регресія;

– градієнтний бустинг (Gradient Boosting, XGBoost, LightGBM);

– випадковий ліс;

– k-найближчих сусідів (k-Nearest Neighbors, k-NN).

Для моделювання складних залежностей використовуються штучні нейронні мережі:

– багатошаровий перцептрон (MLP);

– рекурентні нейронні мережі (RNN);

– довга короткочасна пам’ять (LSTM) для часових рядів;

– графові нейронні мережі (Graph Neural Networks) для транспортних мереж.

Задачі маршрутизації також розв’язуються із застосуванням навчання з підкріпленням, де агент оптимізує політику вибору маршруту на основі функції винагороди.

Програмна реалізація зазначених моделей здійснюється з використанням мов програмування високого рівня, як-от Python, із застосуванням бібліотек:

– NumPy, SciPy – для чисельних обчислень;

– NetworkX – для роботи з графами;

– OR-Tools – для задач оптимізації;

– TensorFlow, PyTorch – для реалізації нейронних мереж;

– scikit-learn – для класичних ML-моделей.

Отже, сучасна транспортна логістика базується на інтеграції математичних моделей, алгоритмів оптимізації та методів штучного інтелекту. Це дає змогу ефективно розв'язувати задачі великої розмірності, підвищувати точність прогнозування та забезпечувати адаптивність логістичних систем до змін зовнішнього середовища.

Список використаних джерел

1. Уманська А. В. Програмний застосунок для управління логістичним обслуговуванням підприємства: дипломна робота. Вінниця, 2026.
2. Toth P., Vigo D. Vehicle Routing: Problems, Methods, and Applications. SIAM, 2014.
3. Laporte G. The Vehicle Routing Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research.* 2009. № 59(3). P. 345–358.
4. Ballou R. H. Business Logistics Management. Pearson, 2004.
5. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016.
6. Scikit-learn: Machine Learning in Python / F. Pedregosa et al. *JMLR.* 2011. № 12. С. 2825–2830.

УДК 004.8

Щербина Д. С., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

МЕТОДИ КЛАСИФІКАЦІЇ СТИЛЮ ШАХОВОГО ГРАВЦЯ ЯК ОСНОВА ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасний розвиток інформаційних технологій, зокрема штучного інтелекту та методів машинного навчання, суттєво трансформує підходи до побудови інтелектуальних систем підтримки прийняття рішень. Одним із ключових напрямів цього розвитку є створення експертних та рекомендаційних систем, здатних адаптуватися до індивідуальних характеристик користувача. У контексті інтелектуальних ігор, як-от шахи, ця проблема набуває особливого значення, оскільки ефективність навчання та вдосконалення ігрових навичок значною мірою залежить від рівня персоналізації навчального процесу. У традиційних підходах до навчання шахам зазвичай використовуються універсальні методики, що не враховують індивідуальні особливості гравця, зокрема його стиль гри, схильність до певних стратегій чи типових рішень. Це знижує ефективність навчання та уповільнює процес досягнення високого рівня майстерності.

Стиль шахового гравця є складною багатовимірною характеристикою, що формується під впливом когнітивних, поведінкових та стратегічних факторів [1]. Він проявляється у виборі дебютів, типових планів у мітельшпілі, півні агресивності, схильності до ризику, здатності до тактичного або позиційного мислення, а також у характері помилок, які допускає гравець. У класичному розумінні виділяють кілька базових стилів, як-от агресивний, позиційний, універсальний

або оборонний, проте в реальних умовах стиль кожного гравця є унікальною комбінацією різних характеристик, що ускладнює завдання його формалізації та подальшого використання в інтелектуальних системах. Саме тому актуальними є завдання розробки ефективних методів класифікації стилю шахового гравця, які можуть бути інтегровані в рекомендаційні системи з метою формування персоналізованих порад щодо вибору дебютів та стратегій гри.

Актуальність цього дослідження обумовлена зростаючим попитом на адаптивні навчальні системи, здатні враховувати індивідуальні особливості користувача. У багатьох сферах, як-от електронна комерція чи мультимедійні сервіси, рекомендаційні системи вже довели свою ефективність, забезпечуючи персоналізований досвід взаємодії [2]. У шахах аналогічний підхід лише починає активно розвиватися, хоча передумови для цього вже сформовані. Зокрема, наявність великих баз шахових партій, можливість отримання детального аналізу позицій за допомогою сучасних рушіїв, а також розвиток онлайн-платформ створюють сприятливе середовище для застосування методів аналізу даних. Водночас важливим є не лише визначення сили гравця, але й глибоке розуміння його стилю, що дає змогу формувати більш релевантні рекомендації. Наприклад, гравцю, який віддає перевагу динамічній і тактичній грі, доцільно рекомендувати дебюти з відкритими позиціями та високою ймовірністю комбінаційної боротьби, тоді як позиційно орієнтованому гравцю більше підійдуть закриті структури з поступовим накопиченням переваги.

Найбільш перспективним напрямом є використання методів машинного навчання, які дають змогу автоматично виявляти складні патерни у даних і будувати більш точні моделі стилю гравця. У межах цього підходу можуть застосовуватися як класичні алгоритми класифікації, наприклад, дерева рішень, випадкові ліси або градієнтний бустинг, так і більш складні моделі, включно з нейронними мережами. Особливий інтерес становить використання методів глибокого навчання для аналізу послідовностей ходів, що дає змогу враховувати контекст гри та динаміку розвитку позиції [3]. До того ж застосування методів зниження розмірності, як-от головні компоненти, дає змогу візуалізувати простір стилів і краще зрозуміти структуру даних.

Ключовим етапом у побудові системи класифікації є формування набору ознак, які найбільш повно відображають стиль гравця. До таких ознак можуть належати характеристики дебютного репертуару, включно з частотою використання відкритих або закритих систем, показники тактичної активності, зокрема кількість комбінацій і жертв, позиційні параметри, як-от контроль центру або структура пішаків, а також поведінкові характеристики, включно з часом на прийняття рішень і стабільність гри. Важливим джерелом інформації є також аналіз помилок, який дає змогу виявити слабкі сторони гравця та врахувати їх під час формування рекомендацій.

Результати класифікації стилю можуть бути ефективно використані у рекомендаційних системах для підвищення якості навчання та прийняття рішень. Зокрема, на основі визначеного стилю система може пропонувати оптимальні дебюти, які відповідають індивідуальним уподобанням гравця, а також рекомендувати стратегії мітельшпілю, що дають змогу максимально реалізувати

його сильні сторони. До того ж можливе формування персоналізованих навчальних програм, що включають задачі та вправи, адаптовані до стилю та рівня користувача. Такий підхід сприяє більш ефективному засвоєнню матеріалу та підвищує мотивацію до навчання.

У сучасних умовах найбільш доцільним є використання гібридних підходів, які поєднують переваги різних методів. Зокрема, експертні правила можуть використовуватися для початкової інтерпретації даних і забезпечення прозорості системи, тоді як методи машинного навчання забезпечують високу точність і здатність до адаптації. Додатково можуть застосовуватися методи колаборативної фільтрації, які дають змогу враховувати досвід інших гравців зі схожими характеристиками, що підвищує якість рекомендацій.

Отже, класифікація стилю шахового гравця є ключовим елементом побудови сучасних рекомендаційних систем у галузі інтелектуальних ігор. Використання сучасних методів аналізу даних дає змогу не лише більш точно визначати індивідуальні особливості гравця, але й ефективно застосовувати ці знання для персоналізації навчального процесу. Запропонований підхід відкриває нові можливості для створення інтелектуальних систем підтримки прийняття рішень у шахах, що відповідають сучасним вимогам до адаптивності та ефективності. У подальших дослідженнях доцільно зосередитися на розширенні набору ознак, удосконаленні моделей машинного навчання, а також інтеграції розроблених рішень у веборієнтовані навчальні платформи, що дасть змогу забезпечити їх практичне застосування та масштабування.

Список використаних джерел

1. Groot A. D. Thought and Choice in Chess. Amsterdam: Mouton Publishers, 2019. 463 p.
2. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 3rd ed. Springer, 2022. 1205 p.
3. McIlroy-Young R., Wang K., Sen S. Learning Personalized Models of Human Behavior in Chess. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2020. P. 1690–1699.

УДК 004.85

Юрчук Д. М., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

ВІДНОВЛЕННЯ ПРОПУСКІВ У МЕТЕОРОЛОГІЧНИХ ЧАСОВИХ РЯДАХ МЕТОДАМИ СТАТИСТИЧНОГО НАВЧАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Метеорологічні спостереження з відкритих джерел, зокрема з глобальної системи NOAA, часто містять пропуски. Причини різні – технічні збої облад-

нання, нерегулярність вимірювань, особливості передачі даних. Класичні методи інтерполяції, як-от лінійна або сплайн-інтерполяція, дають задовільні результати лише для коротких лакун. Коли довжина пропуску сягає кількох діб або більше, прості інтерполяційні підходи стають неефективними, оскільки не враховують багатовимірну структуру метеорологічного процесу [1, с. 207–208].

Альтернативою виступають методи статистичного навчання. Замість механічного з'єднання сусідніх точок вони будують прогностичну модель, яка використовує інформацію з інших метеорологічних показників. Наприклад, пропущене значення температури може бути відновлене на основі атмосферного тиску, швидкості вітру та відносної вологості того самого дня. Серед методів, які добре зарекомендували себе в таких задачах, – ансамблеві підходи на основі дерев рішень. Вони здатні моделювати нелінійні залежності, стійкі до викидів і не потребують жорстких параметричних припущень [2, с. 112–114].

Важливою перевагою цих методів є їх здатність працювати з різнотипними змінними одночасно. Метеорологічний набір даних зазвичай містить як неперервні величини, наприклад, температуру або тиск, так і дискретні, або категоріальні, показники. Ансамблеві методи без додаткових перетворень здатні обробляти таку суміш типів, що робить їх особливо зручними для практичного застосування.

Окрема проблема – коректне навчання таких моделей на часових рядах. Використання стандартної випадкової стратифікації призводить до витоку даних: майбутні спостереження потрапляють у тренувальну вибірку, що дає нереалістично оптимістичну оцінку якості. Тому необхідно застосовувати часову кросвалідацію, коли модель навчається на історичному періоді, а перевіряється на наступному хронологічному інтервалі [3, с. 95–97]. Такий підхід більше відповідає реальним умовам прогнозування, де майбутнє залишається невідомим.

Внаслідок такого підходу вдається досягти значного зниження похибки відновлення, порівняно з класичною інтерполяцією. Найбільший виграш спостерігається для довгих лакун тривалістю від кількох до кількох десятків часових кроків, де лінійні методи практично перестають працювати. До того ж статистичне навчання дає змогу отримати оцінку невизначеності відновлених значень, що є цінною додатковою інформацією для подальшого аналізу. Отже, застосування статистичного навчання дає змогу ефективніше використовувати неповні масиви метеорологічних спостережень.

Список використаних джерел

1. Moritz S., Bartz-Beielstein T. imputeTS: Time Series Missing Value Imputation in R. *The R Journal.* 2017. Vol. 9. № 1. P. 207–218.

2. Stekhoven D. J., Bühlmann P. MissForest – non-parametric missing value imputation for mixed-type data. *Bioinformatics.* 2012. Vol. 28, № 1. P. 112–118.

3. Hyndman R. J., Athanasopoulos G. Forecasting: principles and practice. 2nd ed. OTexts, 2018. 380 p.

УДК 004.8:519.816

Яценко В. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук, старший викладач кафедри інформаційних технологій

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ АГРЕГАЦІЇ У СИСТЕМАХ КОЛЕКТИВНИХ РЕКОМЕНДАЦІЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

Прийняття рішень є ключовим процесом у будь-якій системі, що функціонує в умовах альтернатив і обмежень. Системи підтримки прийняття рішень (СППР) автоматизують цей процес, формуючи рекомендації на основі формалізованих даних про об'єкти вибору та суб'єктів, що приймають рішення [1]. Окремим класом таких систем є рекомендаційні системи – програмні компоненти, що аналізують профілі вподобань користувачів і на їх основі пропонують релевантний контент. Із розвитком цифрових платформ рекомендаційні системи пройшли шлях від простих колаборативних фільтрів до складних моделей на основі нейронних мереж. Проте більшість наявних рішень орієнтовані на індивідуального користувача, тоді як реальні сценарії часто вимагають колективного вибору. Колективне прийняття рішень передбачає визначення єдиної альтернативи для групи учасників і є невід'ємним складником багатьох сфер суспільного життя. Воно застосовується в корпоративному плануванні, медицині, судочинстві, освіті чи в побуті для спільного споживання медіаконтенту. Ключовим етапом тут є агрегація – об'єднання індивідуальних векторів вподобань у єдиний груповий профіль [2]. Попри потенціал, у критичних галузях, як-от медицина (консиліуми) чи судочинство, автоматизовані методи колективної агрегації наразі застосовуються обмежено. Актуальність дослідження зумовлена тим, що впровадження математично обґрунтованих алгоритмів агрегації дає змогу підвищити об'єктивність підсумкового рішення завдяки точному врахуванню думок експертів. Це трансформує процес пошуку консенсусу з суб'єктивного обговорення у верифіковану процедуру, де якість вибору зростає завдяки мінімізації індивідуальної незадоволеності учасників за збереження комфортних умов для більшості групи.

Фундаментальні засади стратегій об'єднання та формалізація критеріїв групового задоволення були закладені ще у 2015 р. J. Masthoff, яка систематизувала підходи до соціальної агрегації [3]. На сучасному етапі, як зазначається у дослідженні X. Cao та Y. Liu (2018), акцент зміщується на використання механізмів уваги для адаптивної агрегації [4]. Такий підхід є прогресивним, оскільки дає змогу системі не просто усереднювати дані, а динамічно визначати «вагу» кожного учасника залежно від контексту ситуації та специфіки об'єктів, що забезпечує потенційно вищу релевантність, порівняно з класичними методами.

Метою дослідження є порівняльний аналіз методів агрегації у системах колективних рекомендацій. Пі час дослідження необхідно описати теоретичне

підґрунтя векторного представлення вподобань та ознак об’єктів, математично формалізувати чотири базові методи агрегації (середнє арифметичне, мінімум, максимум та зважений підхід), провести порівняння їх поведінки за метриками якості та справедливості, запропонувати концепцію комбінованого алгоритму для знаходження оптимального групового компромісу і підсумувати відповідну інформацію.

Для формалізації задачі колективного вибору застосовується підхід, що базується на векторному представленні об’єктів та суб’єктів у єдиному просторі ознак. Векторне представлення – це відображення характеристик об’єкта або профілю користувача у вигляді впорядкованого масиву числових значень. Для фільмів, наприклад, цей багатовимірний простір включає бінарні індикатори жанрів та ключових слів, нормалізовані числові показники середньої оцінки, логарифму популярності, року випуску, тривалості, бюджету, а також нормалізовані показники популярності провідних акторів, що відображають загальну впізнаваність виконавців відповідно до відомих даних. Відповідно на основі історії взаємодій для кожного учасника групи формується індивідуальний вектор уподобань у багатовимірному просторі. Пошук релевантного контенту зводиться до обчислення косинусної схожості між векторами наявних об’єктів та цільовим груповим вектором.

Формування цільового групового вектора потребує застосування відповідного оператора об’єднання. Агрегація векторів – це математична операція зведення множини індивідуальних профілів до єдиного інтегрального представлення. Нехай група G складається з N учасників, а вектор уподобань кожного користувача $\mathrm{u} \in \mathrm{G}$ позначено як $\mathrm{v_u}$. У межах дослідження проаналізовано чотири алгоритми обчислення компонент $v_{G,i}$ цільового групового вектора v_G.

Метод середнього арифметичного обчислює компонентне середнє між усіма векторами учасників:

$$v_{G,i} = \frac{1}{N} \sum_{u \in G} v_{u,i}.$$

Він максимізує сумарну схожість для групи загалом, проте схильний ігнорувати інтереси меншості.

Метод мінімуму для кожного виміру обирає найменше значення серед усіх профілів:

$$v_{G,i} = \min_{u \in G} v_{u,i}.$$

Цей підхід гарантує, що жодна ознака контенту не буде підкреслена понад найнижчий індивідуальний рівень сприйняття. Оператор ефективно запобігає виникненню критичного незадоволення окремих учасників, але суттєво звужує множину потенційних альтернатив, часто зводячи вибір до компромісних, проте малоцікавих об’єктів.

Метод максимуму функціонує за протилежним принципом, обираючи найбільше значення кожної ознаки:

$$v_{G,i} = \max_{u \in G} v_{u,i}.$$

Він спрямований на максимізацію задоволення окремих користувачів шляхом виділення їхніх найсильніших інтересів. Недоліком стратегії є вкрай низька стійкість до полярних або взаємовиключних вподобань усередині колективу.

Зважена агрегація формує груповий вектор шляхом підсумовування з урахуванням коефіцієнтів надійності:

$$v_{G,i} = \frac{\sum_{u \in G} w_u v_{u,i}}{\sum_{u \in G} w_u}.$$

де w_u – вага профілю користувача u, що визначається рівнем його активності (кількістю виставлених раніше оцінок).

Ваги всіх користувачів нормалізовані в такий спосіб, що сума всіх коефіцієнтів дорівнює 1. Профіль із більшим обсягом історичних даних отримує вищу вагу, оскільки точніше та надійніше відображає реальні вподобання. Недоліком методу є можливе ігнорування системою думок користувачів з низькою вагою.

З метою подолання цих обмежень можна застосувати комбіновану стратегію. Ідея полягає у використанні зваженого середнього як базового алгоритму розрахунку з паралельним накладанням жорстких обмежень за методом мінімуму для відсікання критично неприйнятних характеристик. Такий багаторівневий підхід дає змогу зберегти високу загальну точність підбору контенту, одночасно гарантуючи, що жоден із користувачів не отримає свідомо небажаного результату.

Отже, проведений аналіз підтверджує, що вибір методу агрегації є ключовим чинником ефективності систем колективних рекомендацій. Метод середнього арифметичного забезпечує найвищу середню релевантність, але може ігнорувати інтереси окремих учасників. Метод мінімуму гарантує справедливість і реалізує принцип «права вето», проте знижує загальну релевантність рекомендацій. Метод максимуму дає змогу виявляти найбільш виражені інтереси групи, однак результати можуть бути менш збалансованими. Зважена агрегація враховує авторитетність користувачів, але створює ризик домінування окремих профілів. Найбільш перспективним є гібридний підхід, що поєднує переваги зваженої агрегації та принципу мінімуму для досягнення стабільного й справедливого групового компромісу [5].

Список використаних джерел

1. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 3rd ed. Springer, 2022. 1210 p.

2. Aggarwal C. C. Recommender Systems: The Textbook. Springer, 2016. 498 p.

3. Masthoff J. Group Recommender Systems: Aggregation, Satisfaction and Group Attributes. 2015. P. 743–776.

4. Cao X., Liu Y. Attentive Group Recommendation. *SIGIR'18: Proceedings of the 41st International ACM SIGIR Conference.* 2018. P. 645–654.

5. Serbos D., Qi S. Fairness in Target Group Identification. *World Wide Web.* 2022. Vol. 25. P. 1381–1412.

СЕКЦІЯ 5
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.932.2:004.032.26

Бурківський О. С., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Ніколюк П. К., д-р фіз.-мат. наук, професор, професор кафедри інформаційних технологій

РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОЇ ІДЕНТИФІКАЦІЇ ОСІБ ТА ГЛИБОКОГО АНАЛІЗУ ЗОБРАЖЕНЬ ЗА ВІДКРИТИМИ ДАНИМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасна стадія розвитку інформаційного суспільства відзначається стрімким зростанням обсягів візуального контенту. В умовах глобальних криз та інформаційних протистоянь фотографії часто стають інструментом маніпуляцій та поширення дезінформації [1–3]. На цьому тлі надзвичайно гостро постає необхідність швидкої та достовірної верифікації даних. Аналітикам критично важливо оперативно встановити: ким саме є особа на фото, за яких обставин створено світлину, а також яке точне тлумачення мають специфічні деталі (приміром, елементи військового обмундирування, розпізнавальні знаки чи маркування на техніці).

Сучасні методи розвідувальної діяльності з використанням відкритих джерел (OSINT) зазвичай фокусуються на аналізі розосереджених в інтернеті відомостей. Проте в умовах постійного зростання інформаційних потоків ручне опрацювання таких масивів стає вкрай неефективним. Наявні практики ручного реверсивного пошуку зображень вимагають колосальних часових ресурсів, а стандартні пошукові механізми видають масив неструктурованих посилань. Водночас сучасні нейронні мережі стикаються з бар'єрами прямого опрацювання локальних файлів користувача, що створює додаткові технічні перепони для аналітиків. З огляду на це створення програмного комплексу, який поєднує можливості комп'ютерного зору з генеративним штучним інтелектом для оперативного підтвердження візуальних даних, є критично важливим завданням сьогодення.

Метою розробки є побудова інструменту, який зводить до мінімуму ручну працю під час проведення OSINT-розвідок, перетворюючи 20-хвилинний рутинний процес збирання відомостей на дію, що виконується практично миттєво.

Технічно інструмент створено на базі мови Python [4], із залученням бібліотек для мережевої комунікації, формування візуального інтерфейсу та взаємодії з API моделей ШІ (google.generativeai) [5]. Структура системи базується на гібридній схемі і поділяється на три етапи.

Перший етап – «Хмарний Міст» (Cloud Bridge). Оскільки хмарні аналітичні сервіси потребують доступу до графічних даних через публічні URL, програма дає змогу обрати файл із локального сховища та автоматично завантажує його на тимчасово виділений захищений сервер. Формується шлях взаємодії між пристроєм оператора та нейромережами без довготривалого зберігання цифрових даних. Для зручності користувача процес вибору файла реалізовано через

інтуїтивно зрозумілий графічний інтерфейс. Такий підхід додатково гарантує конфіденційність розслідування, адже після завершення аналізу згенероване тимчасове посилання втрачає свою актуальність.

Другий етап – попередня перевірка та накопичення семантичних індикаторів. Програмне забезпечення задіює шлюз SerpApi та алгоритми Google Lens [1] для пошуку «Графа Знань» та візуальних збігів. Якщо особу або об’єкт ідентифіковано, ініціюється запит до Wikipedia API для вилучення довідкових даних, які слугують контекстним ввідним сигналом. Це дає змогу ефективно відфільтрувати інформаційний шум і зосередити увагу алгоритмів на розпізнаних сутностях. Зібрана на цьому етапі базова інформація формує надійний фактологічний фундамент для подальшої інтелектуальної обробки.

Третій етап – деталізований семантичний аналіз за допомогою мультимодальної генеративної архітектури (Google Gemini 2.5 Flash) [6], що становить основну інновацію платформи. Знайдений текстовий фрагмент спільно з візуальним матеріалом надсилається до мовної моделі. Спеціальний запит (промпт) інструктує ШІ діяти як віртуальний слідчий: система верифікує особу, проводить деконструкцію маркерів (елементів гардеробу, розпізнавальних знаків) та формує висновок щодо умов створення кадру. Завдяки поєднанню фотографії із зібраним текстовим контекстом суттєво мінімізується ризик хибних генерацій (так званих «галюцинацій») нейромережі. Кінцевий результат видається оператору у вигляді вичерпного звіту, який не потребує додаткового ручного структурування.

Для демонстрації працездатності програмного забезпечення було здійснено тестовий запуск із застосуванням світлини відомого військового діяча. Процес первинного пошуку інформації в терміналі програми відображено на рис. 1.

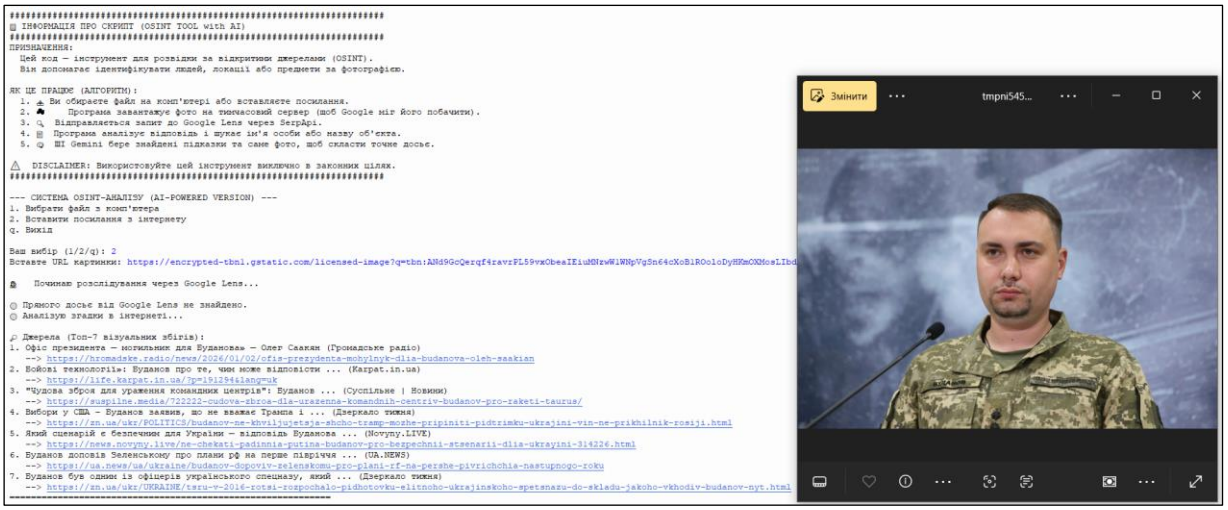


Рисунок 1 – Інтерфейс програми та процес первинної OSINT-ідентифікації

Модуль початкової перевірки успішно впорався із завданням знаходження візуальних відповідностей у загальнодоступних новинах, що дало змогу системі сфокусуватися на вирішальному ідентифікаційному елементі – прізвищі «Буданов». Далі цей ідентифікатор, об’єднаний із вихідним зображенням, був поданий на розгляд моделі штучного інтелекту (рис. 2).

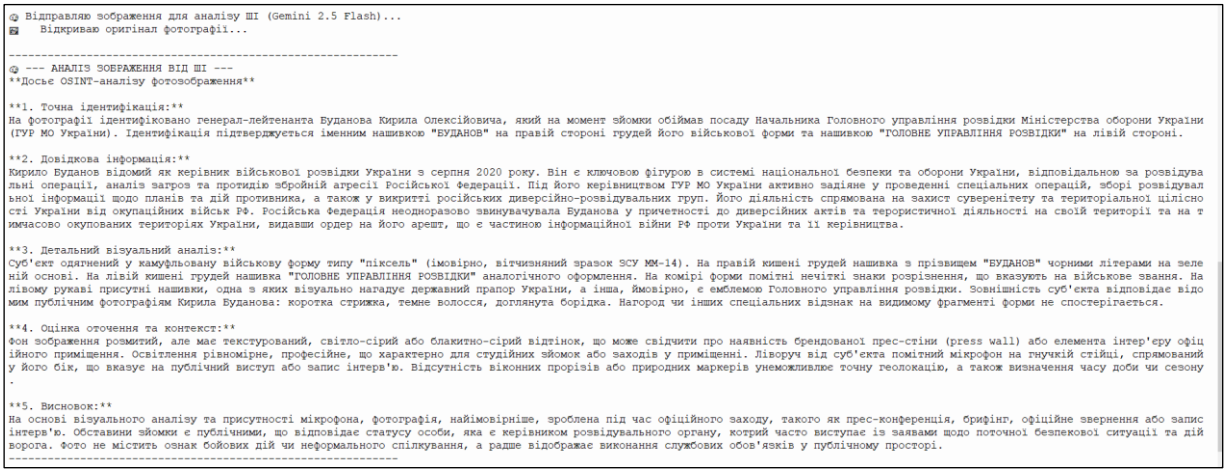


Рисунок 2 – Результат глибокого семантичного аналізу та згенероване ШІ досьє

Як видно з результатів тестування, програмне забезпечення самостійно створило організований аналітичний звіт, провівши ґрунтовний розбір компонентів військового однострою та обставин публічної заяви. Такий структурований формат подачі даних значно полегшує подальшу роботу аналітика.

Підсумовуючи, можна стверджувати, що розроблений програмний комплекс продемонстрував високу ефективність у сфері оперативного аналізу та підтвердження візуальної інформації. Запропонований підхід суттєво оптимізує роботу аналітиків, зменшуючи час на рутинну перевірку фактів. Інструмент успішно обробляє локальні файли користувача та застосовує оригінальне поєднання двох типів нейромереж: пошукової (для швидкого збору фактичних даних) та генеративної (для глибокого аналізу неочевидних візуальних деталей). Внаслідок цього замість масиву непов’язаних посилань користувач миттєво отримує чітко структурований аналітичний звіт-досьє. Проєкт успішно протестований і є повністю готовим до практичного впровадження у щоденну роботу під час OSINT-досліджень та професійного фактчекінгу.

Список використаних джерел

1. Williams H. J., Blum I. Defining Second Generation Open Source Intelligence (OSINT) for the Defense Enterprise. RAND Corporation, 2018. URL: https://www.rand.org/pubs/research_reports/RR1964.html (дата звернення: 30.04.2026).

2. Bazzell M. Open Source Intelligence Techniques: Resources for Searching and Analyzing Online Information. 9th Edition. 2023. URL: https://inteltechniques.com/books.html (дата звернення: 30.04.2026).

3. Hassan N. A., Hijazi R. Open Source Intelligence Methods and Tools: A Practical Guide to Online Intelligence. Apress, 2018. URL: https://link.springer.com/book/10.1007/978-1-4842-3203-3 (дата звернення: 30.04.2026).

4. Python Language Reference, version 3.x. Python Software Foundation. URL: https://www.python.org (дата звернення: 30.04.2026).

5. Google Lens API Documentation. SerpApi. URL: https://serpapi.com/google-lens-api (дата звернення: 30.04.2026).

6. Google Gemini API Documentation. Google for Developers. URL: https://ai.google.dev/docs (дата звернення: 30.04.2026).

УДК 004.418:004.451.9

Гапоянц Д. В., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Фриз І. В., канд. фіз.-мат. наук, старший викладач кафедри інформаційних технологій

АРХІТЕКТУРА СЕРВЕРНОЇ ЧАСТИНИ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ЧИТАННЯ З ІНТЕРАКТИВНИМ СЛОВНИКОМ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні мобільні застосунки для читання поступово перетворюються не лише на засоби перегляду тексту, а й на інтерактивні платформи для навчання та вивчення іноземних мов. Особливо актуальними є застосунки, які дають змогу користувачеві миттєво перекладати слова, зберігати їх у персональному словнику, формувати флешкартки та перевіряти засвоєння нової лексики. У зв'язку з цим важливого значення набуває розробка серверної частини, яка забезпечує зберігання даних, авторизацію користувачів, синхронізацію прогресу читання та взаємодію із зовнішніми сервісами перекладу.

Метою роботи є проєктування архітектури серверної частини мобільного застосунку для читання з інтерактивним словником, який забезпечує керування книгами, переклад слів, збереження словникових записів та підтримку механізмів навчання.

Для реалізації серверної частини використано фреймворк FastAPI, який дає змогу створювати високопродуктивні REST API [1]. Основний модуль застосунку підключає маршрутизатори для роботи з авторизацією, книгами, закладками, прогресом читання, словником, флешкартками та тестуванням. Для роботи з базою даних використано SQLAlchemy та СКБД MySQL, що забезпечує об'єктно-реляційне відображення моделей і централізоване керування підключенням до бази даних [2; 3].

На рис. 1 представлено загальну архітектуру серверної частини мобільного застосунку: мобільний клієнт взаємодіє із сервером через REST API у форматі JSON, FastAPI-сервер забезпечує обробку запитів користувача та взаємодію з базою даних MySQL. Для реалізації перекладу використовується зовнішній сервіс MyMemory API.

Основними сутностями бази даних є користувач, книга, прогрес читання, закладка, словниковий запис, збережене слово та результати тестування. Користувач може додавати книги до власної бібліотеки, зберігати позицію читання та створювати закладки з нотатками. Для цього використовується модель *ReadingProgress,* що містить індекс абзацу та символу, а також модель *Bookmark,* яка дає змогу зберігати позначки у тексті.

На рис. 2 представлено ER-діаграму бази даних серверної частини мобільного застосунку. Основною сутністю системи є користувач, який може створювати власну бібліотеку книг, зберігати прогрес читання, закладки та словникові записи. Для реалізації інтерактивного словника використовуються сутності DICTIONARY_ENTRIES та SAVED_WORDS, що забезпечують збереження перекладів і під-

тримку механізмів повторення слів. Запропонована структура бази даних забезпечує цілісність даних і підтримує основні функції застосунку.

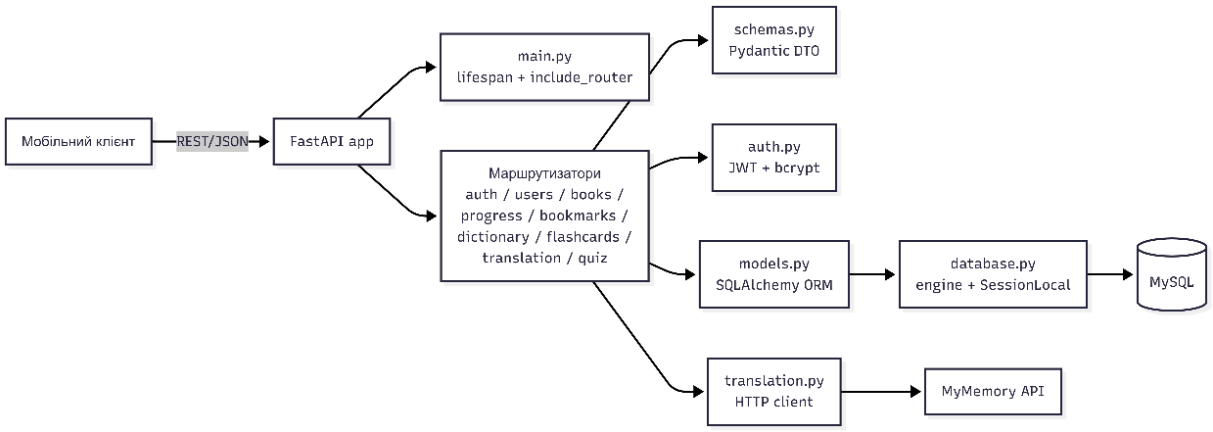


Рисунок 1 – Загальна архітектура серверної частини мобільного застосунку для читання з інтерактивним словником

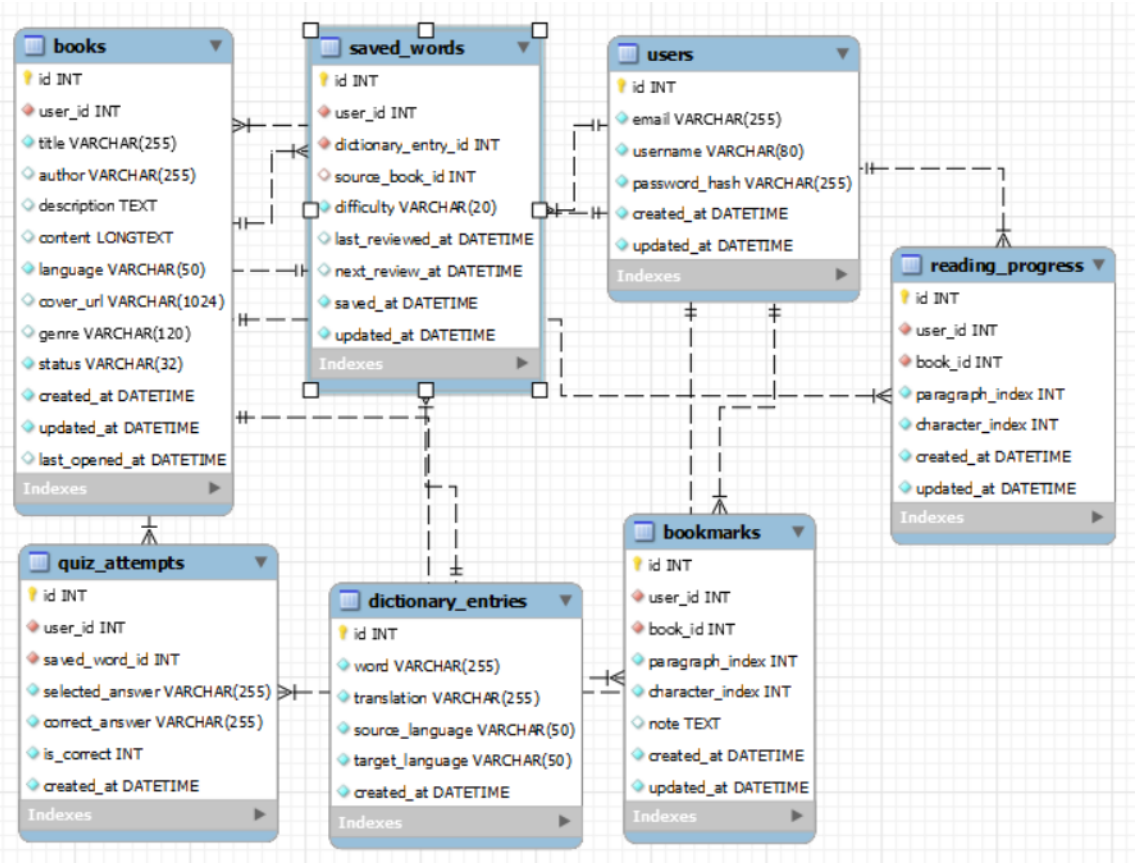


Рисунок 2 – ER-діаграма бази даних серверної частини мобільного застосунку

Особливу роль у системі відіграє інтерактивний словник. Для його реалізації використовуються моделі *DictionaryEntry* та *SavedWord*, які забезпечують збереження слова, перекладу, мовної пари та рівня складності. Також передбачено можливість створення флешкарток та проходження тестування для перевірки засвоєння нової лексики. Модель *QuizAttempt* зберігає інформацію про вибрану користувачем відповідь, правильний варіант та результат перевірки.

Поля difficulty, last_reviewed_at та next_review_at створюють основу для реалізації алгоритмів інтервального повторення (Spaced Repetition System), які широко використовуються у сучасних системах вивчення іноземних мов.

Для забезпечення коректності даних використовуються Pydantic-схеми. Вони описують структуру запитів і відповідей API для реєстрації користувачів, створення книг, перекладу слів, оновлення прогресу читання та роботи зі словником. Це дає змогу виконувати автоматичну валідацію даних і спрощує підтримку серверної частини [5].

Безпека системи реалізована за допомогою JWT-авторизації. У модулі *auth.py* реалізовано механізми хешування паролів із використанням bcrypt, створення access token та перевірки автентичності користувача. Такий підхід забезпечує захист облікових записів та контроль доступу до ресурсів застосунку. Використання JWT-токенів дає змогу реалізувати stateless-автентифікацію, що є поширеним підходом у сучасних REST API та спрощує інтеграцію мобільного клієнта із серверною частиною [4].

Для реалізації функції перекладу використовується окремий модуль translation.py, який взаємодіє із зовнішнім API перекладу через HTTP-запити. У разі недоступності сервісу система повертає повідомлення про помилку перекладу, що підвищує надійність роботи застосунку [6].

Запропонована архітектура серверної частини забезпечує підтримку функцій читання книг, перекладу слів, збереження словникових записів, флешкарток та тестування. Використання FastAPI, SQLAlchemy, MySQL і JWT-авторизації забезпечує модульність, масштабованість та безпечність системи. Розроблену архітектуру можна використовувати як основу для подальшого розвитку мобільного застосунку, зокрема розширення функцій навчання та персоналізації.

Список використаних джерел

1. FastAPI Documentation. URL: https://fastapi.tiangolo.com/ (дата звернення 8.05.2026)
2. SQLAlchemy Documentation. URL: https://www.sqlalchemy.org/ (дата звернення 8.05.2026)
3. MySQL Documentation. URL: https://dev.mysql.com/doc/ (дата звернення 8.05.2026)
4. JWT Introduction. URL: https://jwt.io/introduction/ (дата звернення 8.05.2026)
5. Pydantic Documentation. URL: https://docs.pydantic.dev/ (дата звернення 8.05.2026)
6. HTTPX Documentation. URL: https://www.python-httpx.org/ (дата звернення 8.05.2026)

УДК 004.9:004.421.2

Гуцол Н. А., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ІНФОРМАЦІЙНА СИСТЕМА ОБРОБКИ ІНЦИДЕНТІВ ТА ЗАЯВОК ЗВЕРНЕНЬ КЛІЄНТІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Надання ІТ-послуг неможливе без адекватної технічної підтримки. Без використання спеціальних програм всі звернення від клієнтів зазвичай губляться

у пошті, телефонних дзвінках чи месенджерах. Через цей хаос дедлайни горять, інформація втрачається, а клієнти залишаються незадоволеними.

Тому головною ідеєю є розробка системи Service Desk для збору усіх інцидентів та заявок в одному місці. Це «єдина точка контакту» для клієнтів і відділу сапорту, де все прозоро і зрозуміло. Система реалізована на мові програмування C# та платформі .NET, база даних – на MySQL. Для написання SQL-запитів використано фреймворк – Entity Framework Core з підходом Code-First [2], що дало змогу здійснити опис моделі та автоматичну генерацію таблиць в базі даних. Реалізація фронтенду проведена за допомогою Blazor Server [1] та Tailwind CSS [3]. Інтерфейс головної сторінки системи зі списком заявок представлений на рис. 1.

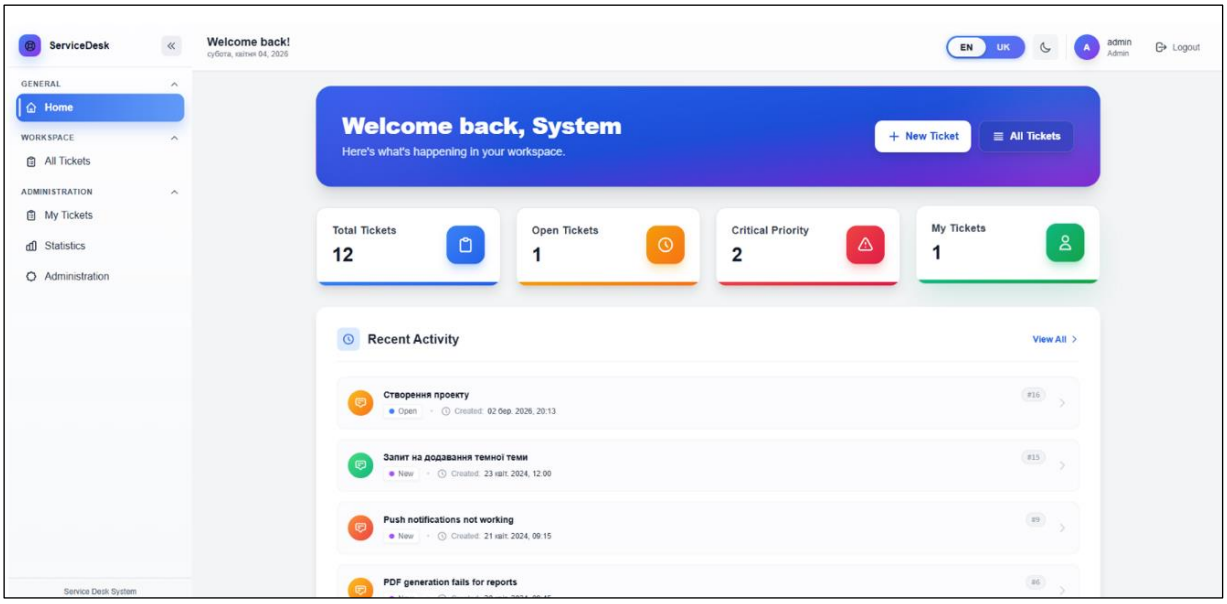


Рисунок 1 – Інтерфейс головної сторінки зі списком заявок

Основу системи складає реалізація загального дашборду. Також у системі створено окрему сторінку для відображення детальної інформації про кожну заявку. На ній можна переглянути повний життєвий цикл інциденту, змінити його статус, а також вести спілкування між користувачем та розробником через систему коментарів, яка оновлюється в реальному часі (рис. 2).

Під час розробки проєкту використано принципи чистої архітектури (Clean Architecture) і реалізовано 5 окремих завдань проєктування: Domain, Application, Infrastructure, Api та Web. База даних складається з 9 логічно пов’язаних таблиць, які розбиті на три блоки: клієнти, продукти та самі тікети. Також є система ролей (Адмін, Розробник, Клієнт), щоб кожна роль працювала у своєму середовищі.

Отже, результатом проєктування та розробки стала інформаційна система Service Desk, яка ефективно вирішує проблематику втрати заявок, спрощує роботу технічної підтримки та має сучасний адаптивний дизайн.

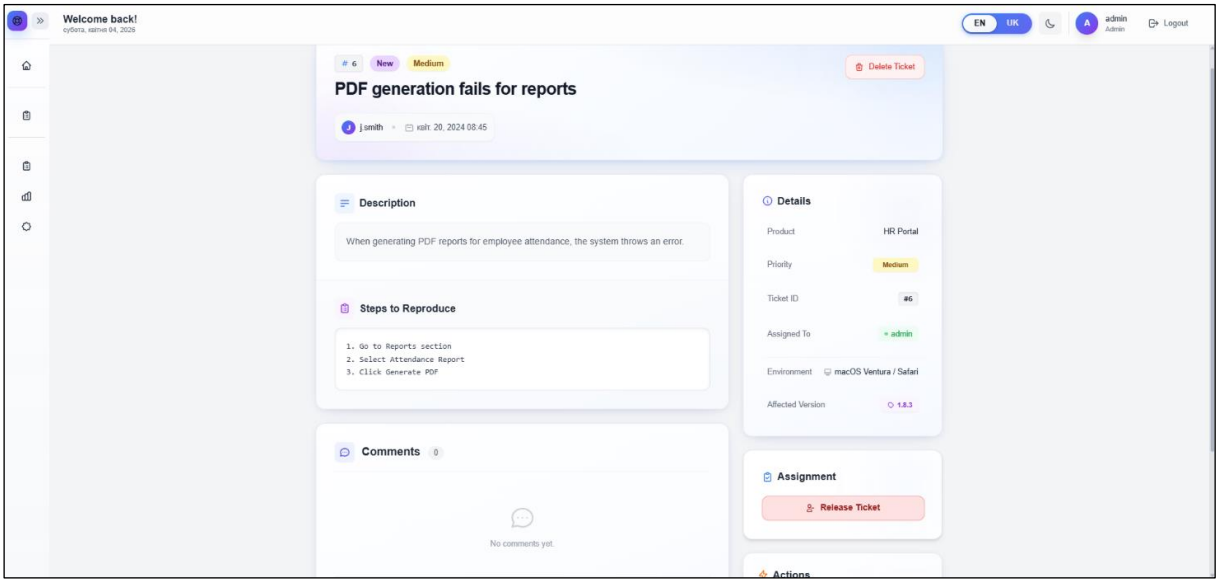


Рисунок 2 – Сторінка деталей заявки та обговорення інциденту

Список використаних джерел

1. Офіційна документація ASP.NET Core Blazor Documentation. URL: https://learn.microsoft.com/en-us/aspnet/core/blazor/
2. Офіційна документація Entity Framework Core. URL: https://learn.microsoft.com/en-us/ef/core/
3. Офіційна документація Tailwind CSS. URL: https://tailwindcss.com/docs

УДК 004.422.5.

Заболотна У. К., здобувачка вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки, Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

ПСИХОФІЗІОЛОГІЯ КОЛЬОРУ В TRAVEL-ІНТЕРФЕЙСАХ: ЯК НЕЙРОМАРКЕТИНГ «ПРОДАЄ» ВІДПУСТКУ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному вебдизайні діє критичне правило «перших 50 мілісекунд»: саме стільки часу потрібно мозку користувача, щоб сформувати підсвідоме ставлення до сайта. У цей мікромомент не задіяне логічне мислення – домінує лімбічна система, що відповідає за емоції та інстинктивні реакції. Колір виступає первинним фільтром сприйняття, що миттєво сигналізує про безпеку, надійність або агресивність цифрового середовища.

Для туристичної сфери вибір палітри безпосередньо впливає на рівень довіри та конверсію. Туристична галузь є однією з найбільш залежних від емоційного стану клієнта – адже рішення про подорож поєднує мрію, фінансовий ризик і довіру до незнайомого сервісу. Блакитний колір у travel-інтерфейсах не просто

виглядає естетично – фізіологічно він знижує частоту серцевих скорочень і стимулює виділення серотоніну, нейромедіатора спокою та безпеки. Сайти з блакитною палітрою демонструють на 15–23 % вищі показники завершення транзакцій, порівняно з ресурсами на нейтральних кольорах.

Керування поглядом відвідувача базується на принципі ізоляції Фон Ресторфф: якщо весь сайт виконаний у спокійних пастельних тонах, мозок переходить у режим релаксації, а раптова поява яскравої кнопки (наприклад, помаранчевої) сприймається як об'єкт найвищого пріоритету. Вибір помаранчевого для заклику до дії (Call-To-Action) має чітке фізіологічне обґрунтування: цей колір посилює постачання кисню до мозку, підвищуючи нейронну активність і готовність до прийняття рішень. Це ідеальний інструмент для розділів із «гарячими турами», де клієнт має діяти швидко та емоційно. A/B-тести провідних travel-платформ підтверджують, що помаранчеві CTA-кнопки підвищують клікабельність на 32–41 %, порівняно зі стандартними рішеннями.

Важливу роль відіграє також концепція білого простору (Negative Space): велика кількість відступів між блоками контенту дає мозку «повітря», знижує когнітивне навантаження та дає змогу користувачу довше залишатися на сайті. Дослідження у галузі UX підтверджують, що грамотне використання негативного простору збільшує середній час перебування на сторінці на 20–35 %, підвищує читабельність тексту та знижує показник відмов. До того ж відкрита композиція формує відчуття преміальності та ексклюзивності бренду.

Психологічні прийоми не працюватимуть без бездоганної технічної реалізації засобами HTML та CSS. Сучасні стандарти CSS надають потужний інструментарій для побудови адаптивних колірних систем. Ключовим механізмом є CSS Custom Properties (змінні), що дають змогу динамічно змінювати «емоційний настрій» усього сайта через модифікацію кількох базових значень. Наприклад, під час переходу до розділу «Гірськолижні курорти» JavaScript може миттєво перемикати клас на кореневому елементі, зміщуючи палітру до холодних відтінків синього та срібного – посилюючи підсвідоме відчуття морозу та адреналіну. Для тропічних напрямків активуються теплі жовто-зелені тони, що асоціюються із сонцем та екзотикою. Технічно це реалізується через ієрархію змінних: глобальні (--color-primary), семантичні (--color-cta) та контекстуальні (--color-winter-bg), що забезпечує одночасно консистентність бренду й гнучкість.

Окрім візуальної привабливості, критичним є дотримання балансу між яскравістю та читабельністю (Contrast Ratio). Стандарти WCAG 2.1 регламентують мінімальний коефіцієнт контрастності між текстом і фоном: не менше 4,5:1 для звичайного тексту та 3:1 для великих заголовків. За даними ВООЗ, приблизно 8 % чоловіків і 0,5 % жінок мають певну форму дальтонізму, тому дизайн, що покладається виключно на колір як носій інформації, є технічно некоректним і юридично проблематичним у багатьох юрисдикціях. Сучасні підходи передбачають використання форми, іконографії та текстових підказок як дублювальних каналів комунікації поряд із кольором. Інструменти автоматизованої перевірки – Lighthouse, Axe, Contrast Checker – дають змогу інтегрувати контроль доступності безпосередньо в процес розробки.

Світові лідери індустрії використовують колір як частину бізнес-стратегії. Booking.com обирає насичений темно-синій (#003580) – колір стабільності та

консервативності, характерний для банків і авіакомпаній, де довіра є фундаментальною цінністю. Натомість Airbnb використовує колір «рауш» (#FF5A5F) – теплий червоно-рожевий відтінок, що асоціюється з домашнім затишком, любов'ю та гостинністю. Жовта гамма Expedia передає радість відкриття, тоді як блакитний Skyscanner апелює до свободи неба та горизонту можливостей.

Дизайн у туристичній сфері – це не лише візуальне оформлення, а наука про те, як сайт змушує клієнта почуватися. Розуміння нейропсихології кольору стає обов'язковою компетенцією вебдизайнера та розробника у висококонкурентному середовищі цифрових сервісів. Перспективними напрямами подальших досліджень є вплив динамічних колірних переходів (CSS Transitions / Animations) на тривалість сесії, а також адаптація колірних стратегій під різні культурні контексти в умовах глобалізації туристичного ринку.

Список використаних джерел

1. Сучасні технології дизайн-діяльності. URL: https://files.znu.edu.ua/files/Bibliobooks/Inshi71/0051457.pdf
2. Доступність для веб-розробників. URL: https://web.dev/articles/a11y-tips-for-web-dev
3. Основи веб-дизайну. URL: https://online.fliphtml5.com/nndfq/lcbx/#p=1

УДК 004.93

Клименко А. Р., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Бацєнко Т. В., канд. техн. наук, доцент, доцент кафедри інформаційних технологій

АНАЛІЗ ВЕБРЕСУРСІВ НА ОСНОВІ ТЕХНОЛОГІЇ ВІДСТЕЖЕННЯ ПОГЛЯДУ КОРИСТУВАЧА

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасній веброзробці важливу роль відіграє не лише функціональність ресурсу, а й зручність його використання. Якість інтерфейсу безпосередньо впливає на швидкість пошуку інформації, ефективність взаємодії та загальний користувацький досвід (UX). Традиційні інструменти аналітики вебресурсів (наприклад, Google Analytics, Hotjar) надають дані про кліки та скролінг, проте не дають змоги точно визначити, на яких елементах користувач концентрує увагу [1].

Одним із перспективних підходів до аналізу поведінки користувача є технологія відстеження погляду (eye-tracking), яка дає змогу визначити напрямок погляду, області концентрації уваги та послідовність перегляду елементів інтерфейсу. Використання стандартних вебкамер у поєднанні з методами комп'ютерного зору робить такі системи доступними для масових досліджень без застосування дорогого спеціалізованого обладнання [2].

Метою роботи є аналіз можливостей використання технології eye-tracking для оцінювання юзабіліті вебресурсів та дослідження поведінки користувачів під час взаємодії з інтерфейсами.

Програмна система має модульну структуру та складається з модуля захоплення відеопотоку, блоку комп'ютерного зору, системи калібрування та модуля аналітики (рис. 1).

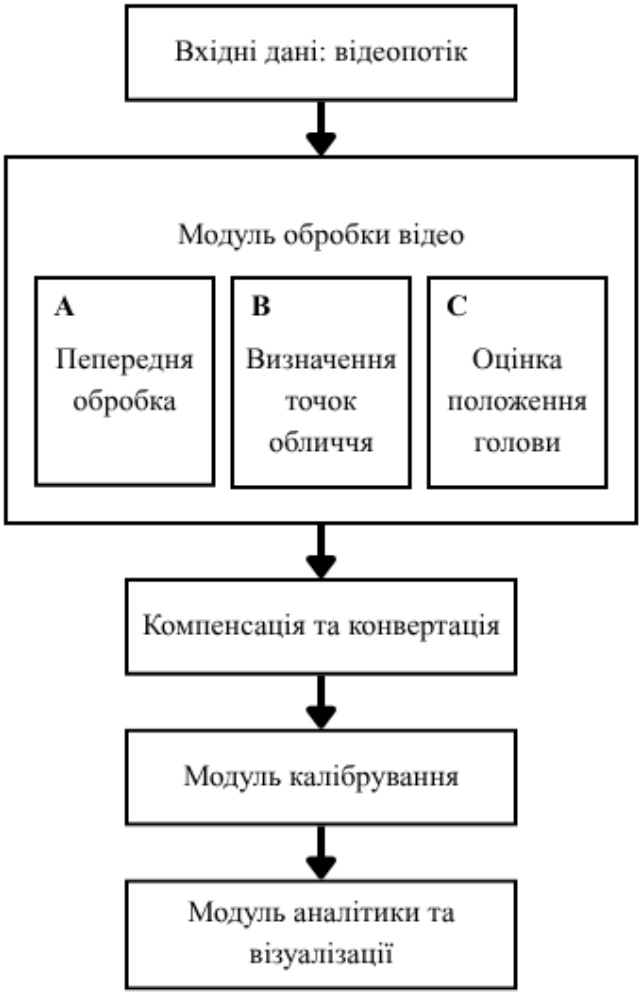


Рисунок 1 – Схема загальної структури програми

Для визначення ключових точок обличчя використовується MediaPipe, а обробка відеопотоку виконується засобами OpenCV. На основі координат очей та визначених точок обличчя знаходиться орієнтований напрямок погляду користувача.

Для підвищення стабільності роботи системи враховується положення голови користувача. Оцінка положення голови здійснюється за допомогою алгоритму cv2.solvePnP, що дає змогу обчислювати кути нахилу та повороту голови. Перед початком роботи виконується перевірка розміщення користувача відносно камери, що дає змогу підвищити точність отриманих даних (алгоритм роботи представлено на рис. 2).

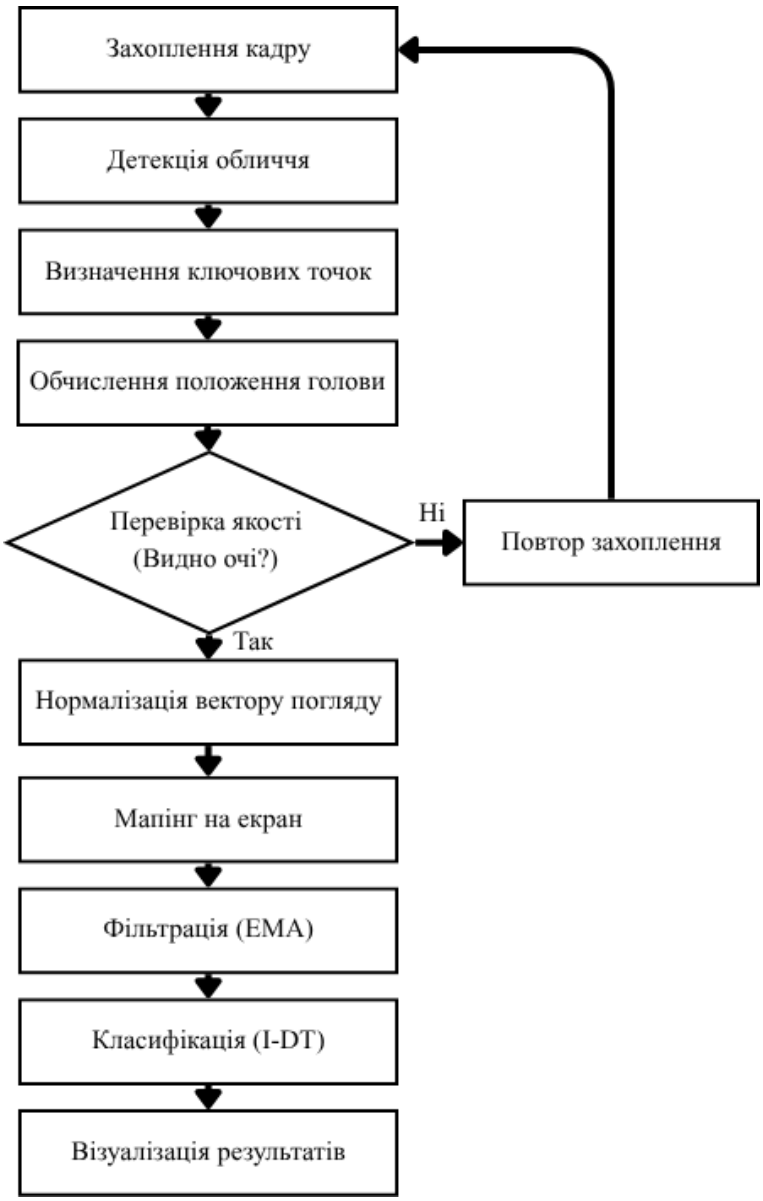


Рисунок 2 – Схема роботи алгоритму eye-tracking

Для перетворення координат із тривимірного простору камери у двовимірну площину екрана застосовується 9-точкова система калібрування. Під час калібрування користувач послідовно фіксує погляд на контрольних точках екрана, після чого система формує відповідність між положенням очей та конкретними пікселями дисплея.

Отримані координати погляду проходять декілька етапів обробки. Для зменшення шумів використовується фільтр експоненціального ковзного середнього (EMA), що забезпечує більш плавне відображення траєкторії погляду.

Для визначення фіксацій застосовується алгоритм I-DT (Dispersion-Threshold), який дає змогу відокремити короткі швидкі рухи очей від ділянок концентрації уваги користувача.

На основі оброблених даних будуються візуалізації:

– Heatmap – теплова карта, що демонструє інтенсивність уваги до різних зон інтерфейсу;

– Gaze Plot – карта траєкторії, яка відображає послідовність та тривалість фіксацій.

Вони дають змогу оцінити ефективність розташування елементів інтерфейсу, структуру сторінки та зручність навігації (рис. 3).

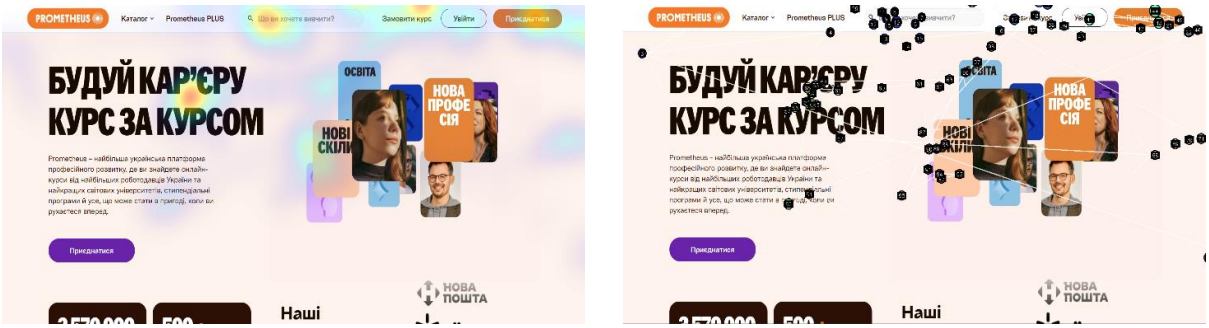


Рисунок 3 – Приклад роботи програми

Для перевірки роботи системи було проведено тестування з різною структурою сторінок та розташуванням елементів навігації.

Дослідження підтвердило гіпотези про «F-патерн та Z-патерн» читання: на одних сторінках основна увага була зосереджена у лівій верхній частині сторінки, а на інших очі рухались від логотипа або заголовка у верхньому лівому куті – до правого краю, далі вниз по центру і знову праворуч. Також було виявлено ефект «банерної сліпоти» – ігнорування елементів, що візуально нагадують рекламу, навіть якщо вони містять корисну навігацію [3; 4].

Основною перевагою запропонованого підходу є можливість використання стандартних камер без необхідності застосування спеціальних інфрачервоних трекерів. А головними обмеженнями є залежність точності від умов освітлення, якість камери та можливі похибки під час використання окулярів або за значних рухів голови.

Результати дослідження показують, що технологія eye-tracking може бути ефективним інструментом для покращення структури вебресурсів, оптимізації навігації та підвищення зручності користувацької взаємодії з вебсайтами.

Список використаних джерел

1. 12 найкращих інструментів веб-аналітики для вашого бізнесу електронної комерції у 2026 р. *eCommerce fastlane.* URL: https://ecommercefastlane.com/uk/11-top-web-analytics-tools-for-your-ecommerce-businesses/ (дата звернення: 05.05.2026).

2. Falkowska J., Sobecki J., Falkowski M. Utilization of eye-tracking metrics to evaluate user experiences – technology description and preliminary study. *Sensors.* 2025. Vol. 25, № 19. P. 6101. URL: https://doi.org/10.3390/s25196101 (date of access: 05.05.2026).

3. Timohin I. Візуальна ієрархія в UI: як дизайн впливає на конверсії. *Pinta.* URL: https://pinta.com.ua/uk/blog/visual-hierarchy-ui-design-user-attention/ (дата звернення: 07.05.2026).

4. Банерна сліпота – причини і наслідки для маркетингу. *PANDA MEDIA.* URL: https://leadpanda.media/blog/banerna-abo-reklamna-slipotaoscho-tse-oznachae/ (дата звернення: 07.05.2026).

5. Кульчицька О. Ю., Січко Т. В. Цифрова обробка зображень та відео. *Прикладні інформаційні технології:* тези доп. всеукр. наук.-практ. конф. (м. Вінниця, 29 квітня 2020 р.). Вінниця, 2020. С. 110–111.

УДК 004.4

Коваленко Д. С., здобувачка вищої освіти 1 курсу спеціальності F3 Комп'ютерні науки,
Бабаков Р. М., д-р техн. наук, доцент, професор кафедри інформаційних технологій

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ БІБЛІОТЕЧНИМ ФОНДОМ НА ОСНОВІ WEB-ТЕХНОЛОГІЙ ТА POSTGRESQL

Донецький національний університет імені Василя Стуса, м. Вінниця

Більшість малих бібліотечних установ в Україні досі покладається на паперовий облік або застаріле програмне забезпечення. Такий підхід робить пошук книг виснажливим, а адміністрування неефективним. Саме тому розробка легкої, кросплатформової інформаційної системи, що працює через браузер, є критично важливою для модернізації освітньої інфраструктури.

Сучасні тенденції розробки програмного забезпечення зміщуються від монолітних структур до архітектурних шаблонів зі зменшеною зв'язністю компонентів системи, як-от Clean Architecture та Vertical Slice Architecture [1]. Окремий пласт досліджень присвячений гнучким архітектурам на базі .NET та Docker, що дають змогу ізолювати бізнес-логіку від інфраструктурних залежностей.

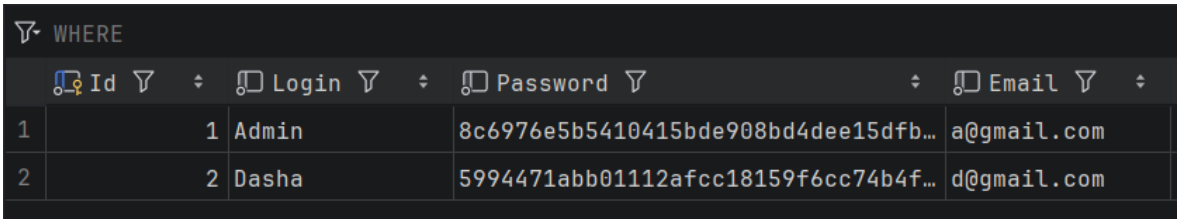
Метою розробки є створення вебзастосунку, який би забезпечив централізоване зберігання даних та рольовий доступ до ресурсів. Для цього розв'язано низку задач: від проєктування реляційної схеми в PostgreSQL [2] до контейнеризації всього стека для швидкого розгортання.

Було обрано інструментами розробки ASP.NET Core та СУБД PostgreSQL. Це поєднання забезпечує високу продуктивність без жодних ліцензійних витрат [1].

Замість звичної багатошарової структури було обрано Vertical Slice Architecture. Такий підхід дав змогу згрупувати код навколо реальних функцій системи: «Каталог», «Читачі», «Видача літератури». Кожен такий «зріз» є незалежним, що значно спрощує навігацію в коді та його тестування за допомогою xUnit.

Робота з даними ведеться через Entity Framework Core за принципом *Code First*, де зв'язки між сутностями (наприклад, між книгою та автором) жорстко контролюються на рівні бази даних [3]. Проєктування об'єктної моделі та взаємодії сутностей базується на принципах об'єктно-орієнтованого програмування [4].

Безпека користувачів реалізована через кастомний AuthMiddleware. Він перевіряє права доступу (RBAC) ще до того, як запит потрапить до бізнес-логіки. Паролі в системі ніколи не зберігаються у відкритому вигляді – для них генеруються незворотні відбитки за алгоритмом SHA-256. Для їх захисту використовується механізм хешування, який перетворює вхідні дані на незворотне значення фіксованої довжини.



WHERE

	Id	Login	Password	Email
1	1	Admin	8c6976e5b5410415bde908bd4dee15dfb…	a@gmail.com
2	2	Dasha	5994471abb01112afcc18159f6cc74b4f…	d@gmail.com

Рисунок 1 – Приклад хешування паролів у системі

Фінальним етапом є підготовка Docker-конфігурації. Вона дає змогу запустити бібліотечну систему на будь-якому сервері за лічені хвилини, уникаючи конфліктів версій програмного забезпечення.

Створена інформаційна система доводить, що навіть за обмежених ресурсів можна побудувати сучасний IT-продукт. Використання модульної архітектури та контейнеризації робить додаток не просто інструментом обліку, а гнучкою платформою, яку легко масштабувати. У майбутньому планується доповнити систему модулем інтелектуальної аналітики для прогнозування попиту на книги.

Список використаних джерел

1. Сидоренко В. В., Константинова Л. В., Смірнов С. А. Організація баз даних: навч. посіб. Кропивницький: ЦНТУ, 2018. URL: https://dspace.kntu.kr.ua/server/api/core/bitstreams/a050035e-e868-4140-b2ba-e449d7394255/content

2. Бублик В. В. Об’єктно-орієнтоване програмування: підручник. Київ: IT-книга, 2015. 624 с. URL: https://itknyga.com.ua/documents/OOP_final.pdf

3. ASP.NET Core Documentation. Microsoft. URL: https://learn.microsoft.com/en-us/aspnet/core/

4. Entity Framework Core Documentation. Microsoft. URL: https://learn.microsoft.com/en-us/ef/core/

УДК 004.855.5:004.514

Костенко Р. О., здобувач вищої освіти 1 курсу ОС «Магістр» спеціальності F3 Комп’ютерні науки,
Ніколюк П. К., д-р фіз.-мат. наук, професор, професор кафедри інформаційних технологій

ПОРІВНЯЛЬНИЙ АНАЛІЗ АСИСТИВНИХ МОБІЛЬНИХ ТЕХНОЛОГІЙ ДЛЯ ОСІБ З ВАДАМИ ЗОРУ: СТАН, ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ

Донецький національний університет імені Василя Стуса, м. Вінниця

За даними Всесвітньої організації охорони здоров’я, понад 2,2 мільярди людей у світі мають порушення зору або сліпоту різного ступеня тяжкості. Водночас більшість із них проживає в країнах із низьким та середнім рівнем доходу, де доступ до спеціалізованих реабілітаційних послуг і дорогих асистивних пристроїв є суттєво обмеженим. Водночас смартфони стрімко набувають

масового поширення навіть у малозабезпечених регіонах – платформа Android займає понад 70 % світового ринку мобільних операційних систем, що робить її природним середовищем для розгортання доступних асистивних рішень. Це формує стійкий запит на комплексні мобільні технології підтримки осіб із вадами зору, що поєднують широкий функціонал із офлайн-доступністю та відсутністю фінансових бар'єрів для користувача.

Сучасний ринок асистивних технологій для незрячих та слабозорих користувачів можна поділити на кілька категорій залежно від форми реалізації та функціонального призначення. Перша категорія – спеціалізовані апаратні пристрої: брайлівські дисплеї, електронні лупи, пристрої для сканування та озвучення тексту. Попри високу ефективність, ці рішення мають суттєвий недолік – значну вартість, яка нерідко перевищує кілька тисяч доларів США, що робить їх практично недоступними для широкого кола користувачів у країнах, що розвиваються [1]. Друга категорія – вбудовані засоби доступності мобільних платформ, зокрема TalkBack для Android та VoiceOver для iOS. Ці інструменти забезпечують базовий скринридерний функціонал і не потребують додаткових витрат, проте їх можливості обмежені переважно озвученням елементів інтерфейсу та не включають розпізнавання навколишнього середовища чи роботу з довільними зображеннями.

Третя та найбільш технологічно розвинена категорія – спеціалізовані мобільні додатки з елементами штучного інтелекту. Серед найвідоміших представників цього напряму варто виділити Microsoft Seeing AI, Google Lookout та Be My Eyes. Seeing AI забезпечує розпізнавання тексту, валют, кольорів, облич та емоцій, однак є доступним виключно для платформи iOS і вимагає постійного підключення до інтернету для більшості функцій [2]. Натомість Google Lookout орієнтований на Android, проте його функціонал залишається обмеженим переважно розпізнаванням документів та об'єктів у режимі реального часу, без повноцінної навігаційної підтримки. Додаток Be My Eyes будується на принципово іншій концепції – залученні волонтерів до живого відеозв'язку з незрячим користувачем, що забезпечує гнучкість, але критично залежить від доступності інтернету та готовності волонтерів.

Порівняльний аналіз зазначених рішень виявляє низку системних прогалин, що залишаються невирішеними в актуальних асистивних технологіях. По-перше, жоден з аналізованих додатків не забезпечує повноцінного офлайн-режиму для всього спектра функцій – залежність від хмарної інфраструктури робить їх непридатними в умовах нестабільного або відсутнього інтернет-з'єднання. По-друге, наявні рішення або є платформо-залежними (лише iOS або лише Android), або вимагають придбання окремих платних підписок для доступу до повного функціоналу. По-третє, більшість наявних додатків спеціалізуються на одній функції – або розпізнаванні тексту, або ідентифікації об'єктів – і не пропонують інтегрованого рішення з єдиним голосовим інтерфейсом керування. Це змушує користувача перемикатися між кількома додатками, що є суттєвою проблемою для осіб із повною відсутністю зору [3].

Поширення відкритих бібліотек машинного навчання, зокрема ML Kit від Google та TensorFlow Lite, суттєво змінює технологічний ландшафт у сфері

мобільних асистивних рішень. Вперше стає можливим запуск складних моделей комп'ютерного зору безпосередньо на мобільному процесорі без необхідності звернення до хмарного сервера. ML Kit забезпечує готові API для оптичного розпізнавання символів, детекції облич і об'єктів, тоді як TensorFlow Lite дає змогу розгортати кастомізовані моделі нейронних мереж із мінімальними вимогами до обчислювальних ресурсів. Ці технології разом відкривають можливість побудови комплексної системи підтримки осіб із вадами зору, яка функціонуватиме на будь-якому Android-пристрої навіть за повної відсутності підключення до мережі.

Окремою дослідницькою проблемою залишається проєктування інтерфейсу взаємодії для користувачів без можливості зорового сприйняття екрана. Традиційні підходи до UX-проєктування, орієнтовані на зрячого користувача, виявляються неефективними в цьому контексті. Голосовий інтерфейс є найбільш природним каналом взаємодії, однак стандартні реалізації мовного розпізнавання зазвичай не враховують контекст поточного стану системи та вимагають точного відтворення фіксованих команд. Перспективним напрямом є розробка контекстно-залежних голосових інтерфейсів, де набір доступних команд динамічно адаптується залежно від активного модуля системи та попередніх дій користувача, що суттєво знижує когнітивне навантаження та підвищує ефективність взаємодії [4].

Отже, проведений аналіз засвідчує, що, попри активний розвиток асистивних мобільних технологій, сьогодні відсутнє комплексне безкоштовне офлайн-рішення для Android, яке поєднувало б у єдиному інтерфейсі оптичне розпізнавання тексту, ідентифікацію об'єктів навколишнього середовища та навігаційну підтримку. Виявлені прогалини обґрунтовують актуальність розробки уніфікованої мобільної системи на основі on-device-методів машинного навчання, орієнтованої на широку аудиторію користувачів із вадами зору незалежно від рівня їхнього доходу та якості інтернет-з'єднання. Подальші дослідження доцільно спрямувати на розробку та практичне тестування архітектури такої системи, оцінку її точності й швидкодії на пристроях різних цінових категорій, а також на дослідження ефективності голосового інтерфейсу в реальних умовах використання.

Список використаних джерел

1. WHO. World report on vision. Geneva: World Health Organization, 2019. 180 p.

2. Szpiro S. F., Zhao Y. Finding a store, searching for a product: a study of daily challenges of low vision people. 2016. P. 61–72.

3. VizWiz Grand Challenge: Answering Visual Questions from Blind People / D. Gurari et al. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018. 10 p.

4. Kane S. K., Bigham J. P., Wobbrock J. O. Slide Rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-touch Interaction Techniques. *Proceedings of ASSETS 2008.* 2008. 8 p.

УДК 004.8

Лисак В. О., здобувач вищої освіти 3 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ НАДАННЯ ПРАВОВОЇ ДОПОМОГИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

В умовах динамічних змін у законодавстві та підвищення потреб суспільства у швидкому доступі до правової інформації, традиційні методи пошуку нормативних актів стають неефективними для користувачів без спеціальної юридичної освіти. Використання публічних великих мовних моделей часто призводить до генерування недостовірної інформації («галюцинацій»).

Актуальність роботи зумовлена необхідністю створення надійного інструменту для автоматизації правових консультацій, який би поєднував точність офіційної бази законодавства зі зручністю інтерфейсу на базі штучного інтелекту.

Сучасні дослідження у сфері великих мовних моделей (LLM) демонструють їх високий потенціал у розумінні природної мови [1]. Проте для предметних областей, що вимагають абсолютної фактологічної точності (як-от юриспруденція), використання базових моделей є ризикованим. Ефективним рішенням цієї проблеми є використання архітектури генерації, доповненої пошуком (Retrieval-Augmented Generation) [2]. Цей підхід обмежує контекст нейромережі лише достовірними джерелами.

Метою дослідження є розробка та впровадження мобільного застосунку «Legal AI Assistant» для надання правової допомоги користувачам на основі інтеграції бази законодавства та великих мовних моделей з використанням архітектури RAG. Реалізація цієї мети дасть змогу створити надійний інструмент, який ефективно вирішує проблему ШІ-галюцинацій, оскільки генеруватиме відповіді виключно на основі чинних офіційних нормативно-правових актів. Для досягнення поставленої мети було визначено такі завдання:

– спроєктувати архітектуру системи [3];
– розробити алгоритми для парсингу та векторизації текстів законів;
– програмно реалізувати серверне API з алгоритмом гібридного пошуку [4];
– створити кросплатформний клієнтський мобільний застосунок з інтуїтивно зрозумілим інтерфейсом [5].

Програмний комплекс спроєктовано на основі архітектурного підходу Clean Architecture, що забезпечило незалежність бізнес-логіки. Серверна частина реалізована на платформі ASP.NET Core (.NET 8). Для збереження бази законодавства використано реляційну СКБД PostgreSQL. Ключовою особливістю стало застосування розширення pgvector, яке дає змогу зберігати семантичні представлення текстів (embeddings), згенеровані за допомогою локальної мовної моделі Llama 3 через інструмент Ollama.

Під час експериментального тестування RAG-конвеєра було виявлено, що суто векторний пошук (на основі косинусної відстані) має знижену ефективність за умови використання неформальної лексики через складну морфологію української мови. Для розв'язання цієї проблеми було розроблено та впроваджено алгоритм гібридного пошуку. Він поєднує семантичний (векторний) та лексичний (за ключовими словами) пошуки із застосуванням математичного методу ранжування Reciprocal Rank Fusion (RRF). Це рішення дало змогу значно підвищити релевантність знайдених законів.

Клієнтський застосунок розроблено за допомогою кросплатформного фреймворка .NET MAUI із використанням патерну MVVM. Інтерфейс системи функціонує у форматі месенджера. Важливим досягненням стала реалізація функції прозорого цитування: згенерована ШІ відповідь супроводжується інтерактивними картками з посиланнями на конкретні статті законів, що підвищує довіру користувача до отриманої консультації.

За результатами роботи створено базовий прототип мобільного застосунку «Legal AI Assistant». Експериментальне тестування підтверджує принципову життєздатність концепції гібридного пошуку по законодавству. Водночас на даному етапі розробки система функціонує з певними обмеженнями, а застосування архітектури RAG лише частково знижує ризик «галюцинацій» мовних моделей під час складних або нетипових запитів. Відповідно подальший розвиток системи першочергово передбачає усунення наявних недоліків шляхом оптимізації алгоритмів пошуку (впровадження стемінгу) та глибокого налаштування системних підказок (prompt engineering) для забезпечення стабільної та високої точності юридичних консультацій.

Список використаних джерел

1. A Survey of Large Language Models / W. X. Zhao, K. Zhou, J. Li et al. *arXiv preprint.* arXiv:2303.18223, 2023. URL: https://arxiv.org/abs/2303.18223

2. Lewis P., Perez E., Piktus A. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems.* 2020. Vol. 33. P. 9459–9474.

3. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. 1st ed. London: Prentice Hall, 2017. 432 p.

4. Розширення pgvector: Open-source vector similarity search for Postgres. GitHub. URL: https://github.com/pgvector/pgvector

5. Офіційна документація .NET MAUI (Multi-platform App UI). Microsoft Learn. URL: https://learn.microsoft.com/uk-ua/dotnet/maui/

УДК 004.4:378

Молодченко Д. В., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

КРОСПЛАТФОРМОВИЙ ЗАСТОСУНОК ДЛЯ ОРГАНІЗАЦІЇ НАВЧАЛЬНОГО ПРОЦЕСУ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному освітньому процесі цифровізація зайшла доволі далеко і відвідувати заняття тепер можна навіть вдома. Але такий прогрес має і свої недоліки. Через велику кількість цифрових сервісів, зосереджених на вирішенні окремих завдань, здобувачі вищої освіти мають справу з великою кількістю потоків інформації. Зменшення кількості джерел надходження інформації значно знизить ймовірність пропуску важливих повідомлень здобувачем. Основною причиною невчасного інформування є відсутність необхідності часто користуватися, наприклад, застосунком корпоративної пошти: користувач не помітив повідомлення, видалив його зі шторки повідомлень разом зі спамом і не знає про існування нового листа.

Проблему використання програми можна вирішити, розробивши застосунок із функціоналом, який користувачу необхідний значно частіше і пов'язаний лише з навчанням. До такого функціоналу входить розклад занять. Перегляд розкладу слугуватиме причиною відкривати застосунок щодня, можливо, навіть кілька разів на день, особливо на початку семестру. Корпоративна пошта слугує засобом комунікації здобувача та викладача або інших працівників навчального закладу. Проте більшість інформації надходить через старосту групи, і для поширення інформації в цьому випадку використовуються месенджери. Це додає корпоративний інформаційний простір у середовище особистої інформації здобувача. Отже, застосунок також повинен містити зручний спосіб донесення інформації для групи людей. Зробити це можна, додавши функціонал чату, в якому писати повідомлення може лише особа з привілейованим доступом, а інші користувачі можуть з'ясовувати деталі в коментарях відповідного повідомлення. Такий підхід створить організований потік інформації.

До систематичних дій здобувачів також входить виконання завдань. Не всі студенти схильні організовувати власний робочий простір, що призводить до втрати файлів. Якщо створити зручний простір для організованого зберігання даних у одному місці, це не лише мінімізує імовірність втрати даних, а й зменшить час виконання роботи та покращить концентрацію уваги на виконанні завдання.

Застосунок для комунікації найкраще створювати для мобільних платформ, але ця програма містить функціонал, пов'язаний з виконанням завдань, а головною платформою для виконання робіт є персональний комп'ютер / ноутбук. Тому для розробки було обрано .NET MAUI [1] для розробки Android- та Windows-версій застосунку. Цей фреймворк не використовується для розробки

програм під Linux, а розробка під MacOS / iOS вимагає наявності пристрою MacOS, саме тому доступ до функціоналу на даних платформах буде здійснений через вебверсію, яка буде розроблена з використанням Blazor Web App [2] та ASP.NET Core. Попри свої недоліки Blazor чудово вписується в Серверна частина буде написана на основі ASP.NET Core Web API [3]. Всі ці інструменти є частиною платформи .NET, яка має потужну базову бібліотеку і дає змогу писати все використовуючи одну мову програмування – C#.

Робота з даними є критично важливою частиною застосунку, і для складних систем важлива надійність і легкість у роботі з базою даних. Таку надійність та легкість надає Entity Framework Core [4]. Він слугує як ORM для взаємодії програми з базою даних і містить багато інструментів для спрощення процесу розробки програмного забезпечення. Прикладом такого спрощення є використання LINQ-запитів у коді для роботи з базою даних. Також Entity Framework Core може змінювати схему бази даних, коли розробник змінює код.

Результатом роботи стане система для зручного планування навчальної діяльності студентів, контролю виконання завдань та взаємодії з іншими студентами в межах одного цифрового простору

Список використаних джерел

1. Microsoft Learn. What is .NET MAUI. URL: https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-10.0 (дата звернення: 09.05.2026)
2. Фурдак В. Що таке .NET Blazor та як його «готувати». URL: https://dou.ua/forums/topic/53130/ (дата звернення: 09.05.2026)
3. Microsoft Learn. Create web APIs with ASP.NET Core. URL: https://learn.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-10.0 (дата звернення: 10.05.2026)
4. Microsoft.EntityFrameworkCore 10.0.7 – NuGet. URL: https://www.nuget.org/packages/microsoft.entityframeworkcore (дата звернення: 10.05.2026)

УДК 004.9

Суліма В. К., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук., доцент, доцент кафедри інформаційних технологій

ТЕХНОЛОГІЇ ЗБОРУ ДАНИХ ІЗ СОЦІАЛЬНИХ МЕРЕЖ ДЛЯ ФОРМУВАННЯ ІНФОРМАЦІЙНИХ ПОТОКІВ АНАЛІТИЧНИХ СИСТЕМ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні соціальні мережі є одним із найбільших джерел даних у цифровому середовищі. Щоденно користувачі створюють значні обсяги інформації у вигляді текстових повідомлень, коментарів, зображень, відео та реакцій. Ці дані становлять цінний ресурс для аналітичних систем, які використовуються у бізнесі, науці, державному управлінні та сфері інформаційної безпеки [1].

У зв'язку з цим важливим є розроблення результативних підходів до збору, передачі та попередньої обробки даних із соціальних мереж.

Основним способом отримання інформації є використання офіційних програмних інтерфейсів (API), які надаються платформами соціальних мереж. API дають змогу отримувати структуровані дані у стандартизованому форматі, що значно спрощує їх подальшу обробку. За допомогою API можна отримувати текст повідомлень, метадані користувачів, часові мітки, інформацію про взаємодії (лайки, репости, коментарі), а також інші параметри. Наприклад, API платформи X (Twitter) дає змогу отримувати стрічку повідомлень у режимі реального часу, що активно використовується в аналітичних системах. Водночас такі інтерфейси мають обмеження, зокрема квоти на кількість запитів, обмежений доступ до персональних даних та політики конфіденційності, які необхідно враховувати під час розробки систем збору даних [1].

Альтернативним підходом є застосування технологій вебскрапінгу. Цей метод передбачає автоматичне отримання даних із вебсторінок шляхом їх аналізу та обробки. Скрапінг дає змогу отримати інформацію, яка недоступна через офіційні API, або розширити обсяг зібраних даних. Однак цей підхід має низку недоліків: залежність від структури вебсторінок, необхідність регулярного оновлення алгоритмів збору, а також потенційні юридичні обмеження [2]. Тому на практиці часто використовують комбінований підхід, що поєднує API та скрапінг.

Важливим елементом сучасних систем є організація збору даних у режимі реального часу. Соціальні мережі генерують інформацію безперервно, тому затримка в обробці може призводити до втрати актуальності результатів аналізу. Для вирішення цього завдання використовуються технології потокової обробки даних. Системи, побудовані на основі брокерів повідомлень, таких як Apache Kafka або RabbitMQ, дають змогу ефективно передавати потоки даних між компонентами аналітичної системи [3]. Наприклад, Kafka може використовуватися для передачі потоків повідомлень від модулів збору до аналітичних компонентів.

Важливим аспектом є архітектура систем збору даних. Зазвичай такі системи будуються за модульним принципом і включають компоненти збору, збереження, обробки та передачі даних. На першому етапі відбувається отримання даних із зовнішніх джерел, після чого вони передаються до сховищ – баз даних або систем зберігання великих даних. Далі здійснюється попередня обробка, яка включає очищення, нормалізацію та структурування інформації [4].

Якість отриманих даних є критично важливим фактором для ефективності аналітичних систем. Дані із соціальних мереж часто містять шум, дублікати, помилки або нерелевантну інформацію. Тому необхідно застосовувати методи очищення та фільтрації даних, які дають змогу підвищити їх достовірність. До таких методів належать видалення дублікатів, обробка пропущених значень, нормалізація тексту та перевірка коректності метаданих [4].

Окрему увагу треба приділити питанням безпеки та конфіденційності. Використання даних соціальних мереж пов'язане з обробкою персональної інформації, що потребує дотримання відповідних правових норм. Зокрема,

необхідно враховувати вимоги регламентів щодо захисту даних, обмежувати доступ до чутливої інформації та застосовувати методи анонімізації [5]. Недотримання цих вимог може призвести до юридичних наслідків та втрати довіри користувачів.

Отже, ефективний збір даних із соціальних мереж потребує поєднання різних технологічних підходів, кожен з яких має свої переваги та обмеження. Використання API забезпечує стабільність і структурованість даних, вебскрапінг розширює можливості доступу до інформації, а потокові технології дають змогу працювати з даними в реальному часі.

Список використаних джерел

1. Russell M. Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and More. 3rd ed. Sebastopol: O'Reilly Media, 2018. 624 p.

2. Mitchell R. Web Scraping with Python: Collecting More Data from the Modern Web. 2nd ed. Sebastopol: O'Reilly Media, 2018. 306 p.

3. Kreps J., Narkhede N., Rao J. Kafka: a distributed messaging system for log processing. *Proceedings of the NetDB Conference.* 2011. URL: https://netman.aiops.org/~peidan/ANM2016/BigDataSystems/ReadingLists/2011NetDB_Kafka_slides.pdf

4. Gandomi A., Haider M. Beyond the hype: big data concepts, methods, and analytics. *International Journal of Information Management.* 2015. Vol. 35, № 2. P. 137–144. DOI: https://doi.org/10.1016/j.ijinfomgt.2014.10.007

5. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation). Official Journal of the European Union. 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj

УДК 004.932:658.562

Курбангалієв Д. М., здобувач вищої освіти 3 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ІНТЕГРАЦІЯ МЕТОДІВ СТАТИСТИЧНОГО НАВЧАННЯ ДЛЯ КОМПЛЕКСНОГО АНАЛІЗУ ДІЯЛЬНОСТІ ІТ-КОМПАНІЙ: ВІД РЕГРЕСІЇ ДО КЛАСТЕРИЗАЦІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

Важливе місце в сучасному управлінні ІТ-бізнесом займає підхід, заснований на даних (data-driven decision making). Оскільки діяльність ІТ-компаній характеризується великими масивами різнорідних метрик (фінансові показники, продуктивність команд, плинність кадрів, задоволеність клієнтів), більшість прикладних аналітичних задач не допускає використання простих евристичних правил або одновимірних оцінок. Саме тому практична ефективність аналітики прямо залежить від обраного математичного апарату. Традиційне використання

виключно методів прогнозування, регресії, або сегментації, кластеризації, окремо одне від одного часто не дає змоги виявити приховані нелінійні залежності. Тому актуальності набирають питання дослідження впливу попередньої сегментації даних на точність і надійність прогнозних регресійних моделей, визначення практичних критеріїв вибору алгоритмів та побудова інтегрованого конвеєра машинного навчання для комплексного аналізу діяльності ІТ-компаній.

Основою для прогнозування кількісних показників, як-от річний прибуток або бюджет проєкту, є регресійний аналіз. Множинна лінійна регресія є базовим методом, який дає змогу оцінити вплив кількох незалежних змінних на цільову метрику. Ітераційна формула для знаходження коефіцієнтів базується на методі найменших квадратів (МНК) і мінімізує суму квадратів залишків:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon, \quad (1)$$

де Y – залежна змінна, наприклад, дохід компанії;

X_i – незалежні предиктори, наприклад, інвестиції в маркетинг, кількість Senior-розробників, витрати на інфраструктуру;

β_j – вільні коефіцієнти моделі,

ε – випадкова похибка [1].

У геометричному змісті методу використовують проведення гіперплощини в n-вимірному просторі ознаки, яка найкраще описує лінійну тенденцію даних. Проте метод характеризується чутливістю до викидів та припускає однорідність дисперсії, що в реальних даних ІТ-сектору, де існують як малі стартапи, так і техногіганти, рідко таке виконується.

Для вирішення цієї проблеми доцільно застосовувати методи навчання без вчителя на етапі попередньої обробки. Алгоритм K-Means є одним із найшвидших алгоритмів для знаходження природних груп, кластерів, у просторі ознак. Цільова функція K-Means полягає в мінімізації внутрішньо кластерної дисперсії:

$$WCSS = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - c_j \right\|^2 \to min, \quad (2)$$

де k – задана кількість кластерів;

c_j – центроїд j-го кластера;

$x_i^{(j)}$ – об’єкт даних, що належить до кластера j [2].

Реалізація комплексного підходу потребує попереднього шкалювання даних, оскільки алгоритми на основі відстаней є чутливими до масштабу змінних. Для гарантованої та стійкої роботи конвеєра мають виконуватися такі умови:

1. Видалення або логарифмування аномальних значень, викидів, у фінансових показниках.

2. Стандартизація даних (Z-score normalization), щоб дисперсія кожної ознаки дорівнювала одиниці, а середнє значення – нулю.

3. Вибір оптимальної кількості кластерів k за допомогою методу «ліктя» (Elbow method) або метрики силуету (Silhouette score).

Чинником, який визначає якість інтегрованої моделі, є гіпотеза про те, що поведінка метрик всередині однорідних груп, наприклад, аутсорсингові компанії малого розміру або продуктові корпорації, є більш лінійною, ніж у загальній генеральній сукупності. За умови правильного виділення кластерів побудова локальних регресійних моделей для кожної групи забезпечує значне зниження середньоквадратичної похибки (MSE).

Розглянемо реалізацію запропонованого підходу у програмуванні на мові Python з використанням бібліотеки scikit-learn. Обчислення спочатку проводиться для глобальної моделі Random Forest Regressor (базова лінія), а потім для ансамблю: «Кластеризація + Регресія». Додатково реалізовано функцію, яка автоматизує процес розбиття та тренування моделей.

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score

# Load and prepare IT company dataset
data = pd.read_csv('it_companies_metrics.csv')
X = data.drop(columns=['Annual_Revenue'])
y = data['Annual_Revenue']

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X_scaled)

# Train a specialized model for each cluster
models = {}
for cluster_id in range(3):
 # Filter data for the current cluster
 X_cluster = X_scaled[clusters == cluster_id]
 y_cluster = y.iloc[clusters == cluster_id]

 # Train random forest regressor
 rf = RandomForestRegressor(n_estimators=100, random_state=42)
 rf.fit(X_cluster, y_cluster)
 models[cluster_id] = rf

 # Evaluate model performance
 y_pred = rf.predict(X_cluster)
 r2 = r2_score(y_cluster, y_pred)
 print(f"Кластер {cluster_id}: R2 Score = {r2:.3f}, Розмір =
{len(X_cluster)}")
```

Для обчислювання було використано набір даних, що містить 1 500 записів про IT-компанії за наступними ознаками: кількість співробітників, витрати на R&D, маркетинговий бюджет, індекс задоволеності клієнтів, частка ринку.

Дані отримано на основі загальних статистичних розподілів, наведених у звітах IT Ukraine Association [3].

Порівняння результатів глобальної регресійної моделі та інтегрованого підходу (кластеризація + локальна регресія) наведено на рис. 1.

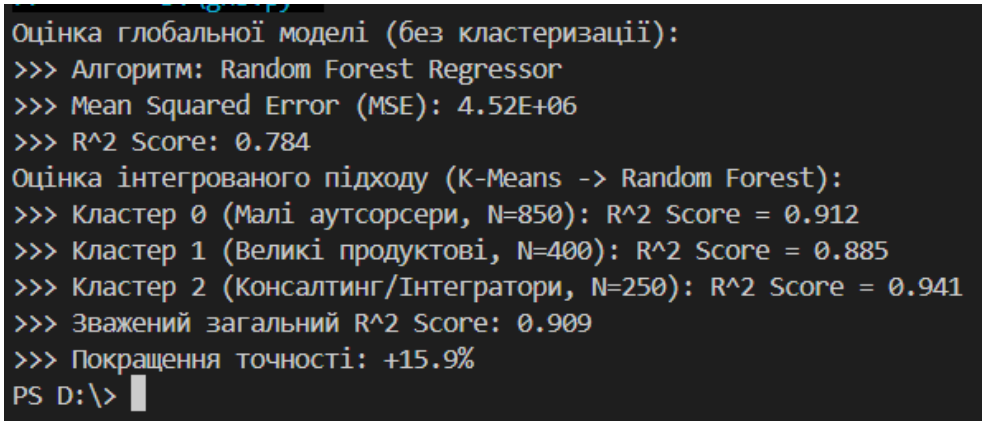


Рисунок 1 – Вивід результатів порівняння глобальної регресійної моделі та інтегрованого підходу (кластеризація + локальна регресія) у консолі

Отже, результати роботи програми підтвердили, що ефективність прогнозування ключових бізнес-метрик IT-компаній суттєво залежить від попереднього відбору ознак. Впровадження автоматизованої кластеризації K-Means перед застосуванням регресійних алгоритмів дає змогу локалізувати складні нелінійні залежності. Результати реалізації коду на мові Python продемонстрували необхідність коректного розбиття на групи: глобальна модель показала точність $R^2 = 0{,}784$, тоді як інтегрований конвеєр допоміг підвищити цей показник до 0,909. Розроблений підхід зменшує зміщення моделі та дає змогу керівникам IT-бізнесу отримувати точніші прогнози, адаптовані до специфіки їх ринкового сегмента.

Список використаних джерел

1. An Introduction to Statistical Learning: with Applications in Python / G. James, D. Witten, T. Hastie, R. Tibshirani. Springer, 2023. 607 p.

2. Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort et al. *Journal of Machine Learning Research.* 2011. Vol. 12. P. 2825–2830.

3. Дослідження IT-галузі України (IT Ukraine Report 2023). Асоціація «IT Ukraine». URL: https://itukraine.org.ua/files/reports/2023/rep2023y_UA.pdf (дата звернення: 01.06.2026).

Наукове видання

Колектив авторів

ПРИКЛАДНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Матеріали

VII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ ТА МОЛОДИХ ВЧЕНИХ

15 травня 2026 року

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

Редактор О. А. Солдатова
Технічний редактор Т. О. Важеніна-Гопрак

Підписано до друку 18.09.2026.
Формат 60×84/16. Папір офсетний.
Друк – цифровий. Умовн. друк. арк. 11,39.
Тираж 30. Зам. 78.

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