

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Всеукраїнська науково-технічна конференція

«Виклики та рішення в програмній інженерії»

26 листопада 2024 року

ЗБІРНИК ТЕЗ



Київ 2024

Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії». Збірник тез. – К.: ДУІКТ, 2024, 224 с.

Даний збірник містить тези учасників конференції, представлених на Всеукраїнській науково-технічній конференції «Виклики та рішення в програмній інженерії», яка проводилась 26 листопада 2024 року на кафедрі Інженерії програмного забезпечення Навчально-наукового інституту інформаційних технологій Державного університету інформаційно-комунікаційних технологій, м. Київ.

Робоча мова конференції – українська та англійська.

У збірнику представлено тези доповідей Всеукраїнської науково-технічної конференції «Виклики та рішення в програмній інженерії», які висвітлюють сучасні тенденції, інноваційні підходи та результати досліджень у сфері програмної інженерії, зокрема розробки, тестування, впровадження та оптимізації програмного забезпечення.

ОРГАНІЗАТОРИ КОНФЕРЕНЦІЇ

Державний університет інформаційно-комунікаційних технологій
Навчально-науковий інститут інформаційних технологій
Кафедра Інженерії програмного забезпечення

ПРОГРАМНИЙ КОМІТЕТ

Бондарчук А.П. – д.т.н., професор, Державний університет інформаційно-комунікаційних технологій

Замрій І.В. – д.т.н., доцент, Державний університет інформаційно-комунікаційних технологій

Барабаш О.В. – д.т.н., професор, Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

Іларіонов О.Є. – к.т.н., доцент, Київський національний університет імені Тараса Шевченка

Золотухіна О.А. – к.т.н., доцент, Київський національний університет імені Тараса Шевченка

Задонцев Ю.В. – к.т.н., Державний університет інформаційно-комунікаційних технологій

Залива В.В. – доктор філософії (PhD), Державний університет інформаційно-комунікаційних технологій

Садовенко В.С. – к.ф.-м.н., доцент, Державний університет інформаційно-комунікаційних технологій

Худік Б.О. – доктор філософії (PhD), Державний університет інформаційно-комунікаційних технологій

Щербина І.С. – к.т.н., доцент, Державний університет інформаційно-комунікаційних технологій

Шевченко С.М. – к.п.н., доцент, Київський столичний університет імені Бориса Грінченка

Яскевич О.В. – к.т.н., Київський столичний університет імені Бориса Грінченка

ЗМІСТ

Кутняк М.Ю., Яскевич В.О.	
Порівняльний аналіз популярних фреймворків Java	8
Адамович В.О., Довженко Т.П.	
Розробка веб-додатку на Python та JS для управління проєктами та відстеження виконання завдань	11
Василець Д.Б., Соляник Л.О.	
Використання SSL/TLS для забезпечення конфіденційності і цілісності даних користувача при роботі з зовнішніми ресурсами в мережі	15
Агашков А.Ю., Шевченко С.М.	
Використання Amazon Web Services у логістиці вантажопасажирських перевезень	19
Андрюшин О.О., Худік Б.О.	
Методика формалізації пошукового запиту на основі методів обробки природньої мови	22
Бенедюк Д.В., Щербина І.С.	
Розробка методу автоматичного аналізу великих обсягів текстової інформації за допомогою ASP.NET	23
Василець Д.Б., Соляник Л.О.	
Визначення вимог до web-застосунку для пошуку наукових публікацій за ключовими словами	25
Власенко І.Ю., Залива В.В.	
Адаптивне кешування даних для підвищення ефективності односторінкових веб-додатків	28
Гапай М.Ю., Фесенко М.А.	
Огляд особливостей мультиголосового поділу для схожих голосів	30
Жужков Д.І., Бондарчук А.П.	
Методика синтезу шрифту за зразком на основі методів розпізнавання зображень	32
Рудковський М.О., Шевченко С.М.	
Покращення продуктивності web-застосунку підтримки клієнтів біржі	37
Соломоник О.П., Компан С.В.	
Інтеграція потокової та історичної обробки даних для інтелектуального аналізу	39
Чернявський Ж.А., Поперешняк С.В.	
Методи та засоби сканування місцевості та візуалізації даних ультразвукового сенсора на основі ехолокації	43
Яценко Д.В., Садовенко В.С.	
Порівняльний аналіз систем предиктивної аналітики	45
Гашко А.О., Бондарчук А.П.	
Визначення вимог до паспортизації діджитал продуктів на основі блокчейну	51
Побережник А.А., Соляник Л.О.	
Порівняльний аналіз продуктивності ORM-технологій для типових CRUD-операцій у веб-застосунках на основі C# та ASP.NET	53
Шульга С.С., Худік Б.О.	
Розробка методу лексичної валідації та корекції слів на основі словника у текстах українською мовою	55
Шульга С.С., Худік Б.О.	
Алгоритм аналізу слів за допомогою словника в системі корекції текстів українською мовою	57
Бенедюк Д.В., Щербина І.С.	

Архітектура системи автоматичного аналізу текстової інформації на базі ASP.NET	60
Бажан Ю.П., Золотухіна О.А.	
Аналіз недоліків автоматизованого тестування UX/UI дизайну на основі методів штучного інтелекту	62
Безугла В.Ю., Щербина І.С.	
Техніки тест-дизайну для підвищення ефективності тестування програмного забезпечення	64
Білодід Д.В., Шахматов І.О., Довженко Т.П.	
Захист фронтенд-компонентів від XSS-атак за допомогою Content Security Policy	67
Кулаков О.В., Замрій І.В.	
Інтеграція метрик відстежуваності та поведінкових метрик для оптимізації рефакторингових рішень у програмних системах	71
Фетько Ю.І., Залива В.В.	
Оптимізація процесу розробки програмного забезпечення шляхом впровадження алгоритму інкрементної обробки файлів	78
Шевчук Ю.О., Замрій І.В.	
Адаптивні алгоритми перевірки знань у вивченні мови програмування JAVA	80
Шоробура В.В., Худік Б.О.	
Розробка методу оптимізації програмного коду на основі генетичного алгоритму	82
Юхта М.А., Компан С.В.	
Методика UX-оптимізації веб-ресурсів для підвищення залученості користувачів в ігрових онлайн-проектах	86
Скидан П.В., Замрій І.В.	
Врахування ризиків та адаптивності при визначенні ефективності моделі прогнозування поведінки фінансових ринків	88
Білодід Д.В., Довженко Т.П.	
Ефективність CSRF-токенів у запобіганні міжсайтовим запитам у фронтенд-додатках	92
Юхта М.А., Компан С.В.	
Аналіз ефективності ux-оптимізації в ігрових проектах на основі інформаційних систем аналітики поведінки користувачів	97
Дзядевич Д.Д., Садовенко В.С.	
Аналіз і прогнозування ризиків у стартапах за допомогою сучасних ML-алгоритмів	100
Кутняк М.Ю., Яскевич В.О.	
Аналіз алгоритму «Зворотного поширення помилки» для навчання нейронних мереж	102
Лабун Є.М., Золотухіна О.А.	
Використання нейронних мереж для автоматизації мастерингу аудіо	105
Левченко О.О., Довженко Т.П.	
Система обробки голосових повідомлень для гарячої лінії на основі методів машинного навчання	109
Лукашенко І.П., Корецька В.О.	
Визначення вимог до експертної системи для підтримки прийняття рішень на основі штучного інтелекту для лікарів первинної медицини	113
Миколаєнко В.О., Сторчак К.П.	
Business Logic Generation using AI technology	117
Пальчук М.Е., Корецька В.О.	
Адаптивне оновлення моделей розпізнавання дорожніх знаків на мобільних пристроях	121
Петрунчак А.Р., Довженко Т.П.	

Оптимізація процесу автоматизованого тестування за допомогою впровадження технології GitHub copilot	123
Решетнік Н.О., Трінтіна Н.А.	
Використання Semantic Kernel для оркестрації функціоналу LLM в розробці ПЗ	125
Соломоник О.П., Компан С.В.	
Розробка системи машинного навчання для оптимізації банківських операцій	129
Спіцин А.Я., Садовенко В.С.	
Використання моделі на основі напівконтрольованого машинного навчання для визначення пріоритетності пацієнтів лікарні	132
Товстенко М.А., Садовенко В.С.	
Візуалізація багатомірної векторної бази даних	134
Федоренко М.Л., Замрій І.В.	
Штучний інтелект як фінансовий порадник: подолання бар'єрів довіри та адаптація до потреб користувача	137
Юр'єв А.Л., Сватко В.В.	
Автоматизоване машинне навчання: прийняття рішень на основі штучного інтелекту у бізнес аналітиці	140
Побережник А.А., Соляник Л.О.	
Метод аналізу продуктивності ORM-технологій у веб-застосунках на С# з використанням ASP.NET: Порівняльне дослідження на основі типових операцій з базою даних	144
Лукашенко І.П., Корецька В.О.	
Аналіз сучасних засобів програмної інженерії для розробки експертних систем	146
Петрунчак А.Р., Шахматов І.О., Довженко Т.П.	
Алгоритм впровадження шучного інтелекту у процес написання коду на прикладі chatgpt	149
Спіцин А.Я., Садовенко В.С.	
Визначення пріоритетності пацієнтів у відділенні невідкладної допомоги як задача машинного навчання	151
Левченко О.О., Довженко Т.П.	
Аналіз ефективності використання нейронних мереж для розпізнавання емоцій в голосових повідомленнях	154
Дзядевич Д.Д., Садовенко В.С.	
Автоматизація оцінки ризиків стартапів за допомогою методів штучного інтелекту	157
Бойко М.С., Бондарчук А.П.	
Застосування шаблонів проєктування Factory та Observer у рушії Unity для створення інвентарю гравця	159
Косенко Д.М., Гребенюк В.В.	
Застосування методів процедурної генерації рівнів для ігор жанру Tower Defence	162
Левчук М.П., Довженко Т.П.	
Роль штучного інтелекту у розвитку сучасних відеоігор	165
Линник Я.А., Довженко Т.П.	
Визначення вимог до організації роботи додатку з користувачем у контексті доповненої реальності	168
Миколаєнко В.О., Сторчак К.П.	
User Interface Generation using AI technology	170
Юхта М.А., Компан С.В.	
Інтеграція сучасних інформаційних технологій у процес UX-оптимізації веб-платформ ігрових спільнот	174

Яковчук В.А., Залива В.В.	
Використання LOD-моделей адаптивного освітлення для оптимізації рендерингу	176
Ярошевський О.В., Замрій І.В.	
Динамічне прийняття рішень на основі особистісних рис ігрових персонажів	178
Бойко М.С., Бондарчук А.П.	
Використання та редагування базового алгоритму псевдовипадкових чисел C# для генерації подій у грі	181
Косенко Д.М., Гребенюк В.В.	
Визначення методів процедурної генерації рівнів для ігор жанру Tower Defence	184
Линник Я.А., Довженко Т.П.	
Огляд технологій доповненої реальності для візуалізації військової техніки та військових технологій в ігровому рушії unity	187
Левчук М.П., Довженко Т.П.	
Огляд методів провадження штучного інтелекту в гру на ігровому рушії Unity	190
Яковчук В.А., Залива В.В.	
Використання рендерингу з багатозональною роздільною здатністю для підвищення продуктивності	193
Ярошевський О.В., Замрій І.В.	
Теоретичні основи формування особистісних рис ігрових персонажів	196
Власенко І.Ю., Залива В.В.	
Оптимізація керування даними в односторінкових веб-додатках через адаптивне кешування	200
Златоус О.Д., Задонцев Ю.В.	
Огляд розробки алгоритмів для автоматичного виявлення та оцінки ризиків у проектному управлінні	203
Колодюк А.В., Жебка В.В.	
Розробка чат-боту для автоматизації клієнтської підтримки у сфері малого та середнього бізнесу	209
Фетько Ю.І., Залива В.В.	
Підвищення ефективності розробки програмного забезпечення шляхом оптимізації процесів обробки ресурсів	211
Шкапа В.В., Замрій І.В.	
Гріді-алгоритми на класах ψ, β-диференційовних періодичних функцій	214
Коваленко Д.С., Лашевська Н.О.	
Концептуальна та математична модель адаптивного алгоритму маршрутизації для С2С логістики із застосуванням глибокого навчання	216
Златоус О.Д., Задонцев Ю.В.	
Розробка методик та засобів підтримки прийняття рішень для управління проектами на основі аналізу ризиків	219
Безугла В.Ю., Щербина І.С.	
Огляд дизайн технік, заснованих на досвіді, які використовуються при тесуванні продуктів програмного забезпечення	221

СЕКЦІЯ 1. НОВІТНІ ПАРАДИГМИ В ПРОГРАМНІЙ ІНЖЕНЕРІЇ

Порівняльний аналіз популярних фреймворків Java

*Кутняк Максим Юрійович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
maxim.kutnyak@gmail.com*

*Науковий керівник: Яскевич Владислав Олександрович,
к.т.н., доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

На сьогоднішній день технології розвиваються дуже швидко та час життя проекту зменшується. Щоб встигати за цим темпом були розроблено велику кількість фреймворків. Наразі вони існують під всі поширені мови програмування та мають як свої переваги, так і недоліки.

Постановка задачі

Поставленою задачею є дослідження роботи найпоширеніших фреймворків на мові Java.

Мета дослідження

Метою даного дослідження є визначення слабких та сильних сторін найпопулярніших фреймворків на мові Java.

Результат дослідження

1. Spring Framework

Spring Framework — це один з найпоширеніших, легких інфраструктур програмного забезпечення, створений для проектування, розробки та розгортання програмного забезпечення на Java [1].

Переваги Spring:

- Можливість запускати веб-додатки навіть без необхідності фактичного веб-сервера,

- Spring здатний тестувати додаток разом із зворотною сумісністю,
- Spring сумісний із конфігураціями XML,
- Spring здатний встановлювати підключення JDBC,
- Spring робить програми схильними до меншої кількості помилок і таким чином підвищує надійність.

Недоліки Spring:

- Spring може бути складним для налаштування, особливо для новачків, через велику кількість опцій і конфігурацій,
- Для нових користувачів і навіть для досвідчених розробників, які переходять з інших фреймворків, крива навчання може бути досить крутою,
- Хоча Spring Boot спрощує конфігурацію, класичний Spring часто вимагає написання великої кількості коду для налаштування контексту, конфігурації та залежностей,
- В деяких випадках, особливо при неправильній конфігурації, Spring може мати продуктивні проблеми, такі як повільний старт або підвищене використання ресурсів,
- Spring часто використовує велику кількість сторонніх бібліотек, що може створити додаткові залежності та потенційні конфлікти версій,

2. Hibernate

Hibernate —фреймворк, який здатен розширити підтримку Java Persistence API [2].

Переваги Hibernate [3]:

- Hibernate є повністю відкритим кодом,
- Hibernate зменшує надмірність через JDBC API,
- Hibernate покращує продуктивність і ремонтпридатність,
- Hibernate підтримує Persistence API,
- ORM Hibernate дозволяє спілкуватися між програмою та будь-якою базою даних.

Недоліки Hibernate [3]:

- Налаштування Hibernate може бути досить складним і вимагати значного часу, особливо для новачків,
- Hibernate може створювати високі накладні витрати на продуктивність, особливо при обробці великих обсягів даних або складних запитів,
- Керування транзакціями в Hibernate може бути заплутаним і складним, особливо у великих додатках з багатьма одночасними операціями,
- Hibernate автоматично генерує SQL-запити, і іноді ці запити можуть бути занадто складними або неефективними,
- Hibernate може неефективно використовувати ресурси, що може призвести до підвищеного споживання пам'яті і ресурсів процесора.

Висновки та перспективи

В тезі було розглянуто 3 найпопулярніших фреймворка на Java. Зокрема Spring, Hibernate, Apache Struts.

Рекомендується використовувати Spring для повного написання веб додатків або веб модулів. Hibernate для прискорення роботи з базою даних, пришвидшенню написання SQL-запитів. Apache Struts можна використовувати для розробки веб додатків, але через свою досить низьку гнучкість, проблеми з безпекою та відсталість від Spring, краще використовувати Spring для цих задач.

Список використаних джерел:

1. Spring Boot. *Spring Boot*. URL: <https://spring.io/projects/spring-boot> (дата звернення: 24.11.2024).
2. Hibernate. Everything data. *Hibernate*. URL: <https://hibernate.org/> (дата звернення: 24.11.2024).
3. Team A. 10 most popular java web development framework in 2024 - aimprosoft - aimprosoft. *Aimprosoft*. URL: <https://www.aimprosoft.com/blog/java-framework-for-web-development/> (дата звернення: 17.05.2024).

Розробка веб-додатку на Python та JS для управління проєктами та відстеження виконання завдань

*Адамович Владислав Олегович,
студент групи ПД-43,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
st7542113@stud.duikt.edu.ua*

*Шахматов Іван Олександрович
Викладач кафедри ПЗ
Державного університету інформаційно-комунікаційних технологій
ivan.shakhmatov@gmail.com*

*Науковий керівник: Довженко Тимур Павлович,
к.т.н., доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Сучасний процес управління проєктами вимагає високого рівня організації та автоматизації. Веб-додатки, розроблені за допомогою технологій Python та JavaScript, є одним із найефективніших інструментів для реалізації таких задач. Зокрема, впровадження веб-додатку для управління проєктами та відстеження виконання завдань дозволяє оптимізувати комунікацію між учасниками команди, полегшити управління ресурсами та покращити контроль за виконанням задач [3-4].

Постановка задачі.

Задачею даного дослідження є визначення вимог до функціональних можливостей та архітектури такого веб-додатку, а також вибір ефективних засобів реалізації. Основна увага приділяється інтеграції ключових функцій, таких як створення та управління проєктами, призначення та моніторинг виконання завдань учасниками команди, аналіз ризиків, а також надання рекомендацій для підвищення ефективності роботи команди. Дослідження спрямоване на розробку рішення, яке покращить комунікацію між учасниками, оптимізує управління ресурсами та забезпечить прозорість процесів у межах проєкту.

Мета дослідження.

Метою дослідження є розробка веб-додатку для управління проєктами та відстеження виконання завдань на основі сучасних технологій Python та JavaScript.

Це включає визначення ключових функціональних можливостей та оптимальної архітектури додатку, а також вибір ефективних інструментів та методів реалізації для створення зручного та інтуїтивно зрозумілого продукту. Додаток повинен забезпечувати можливості створення проєктів, призначення завдань учасникам команди, аналізу ризиків та надання рекомендацій для підвищення ефективності роботи команди. Дослідження спрямоване на оптимізацію процесів управління проєктами та покращення комунікації між учасниками за допомогою інтегрованого програмного рішення.

Результати дослідження.

На основі визначеної архітектури було розроблено діаграму послідовностей, яка ілюструє процеси взаємодії користувача із системою. Діаграма відображає ключові кроки, такі як вхід у систему, створення проєктів, призначення завдань та отримання рекомендацій з аналізу ризиків (Рис. 1).

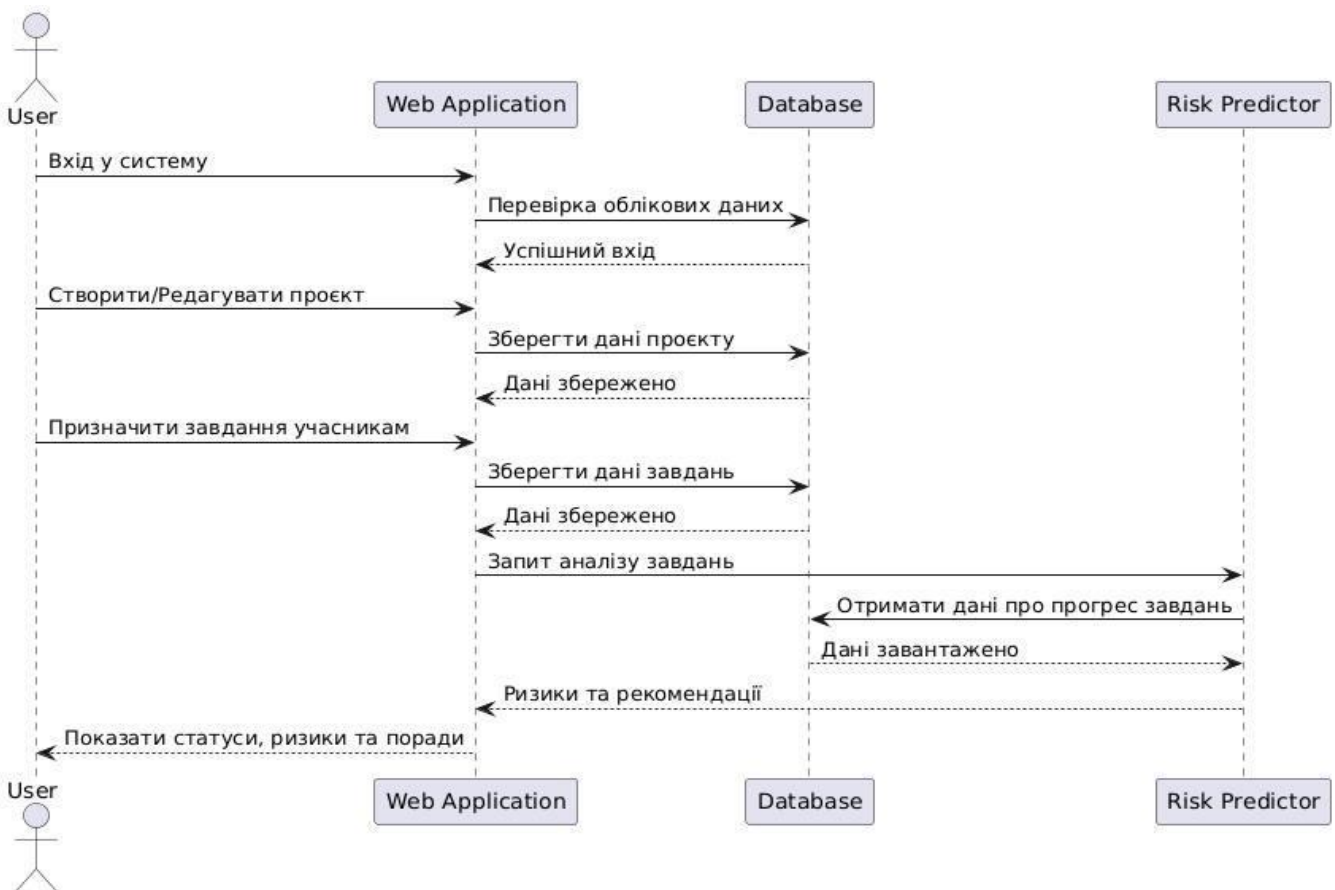


Рис. 1. Діаграма послідовностей

Робота додатку базується на алгоритмі, який забезпечує послідовність виконання операцій, починаючи з перевірки прав доступу до внесення даних у базу та генерації рекомендацій. Алгоритм дозволяє звести взаємодію користувача з додатком до інтуїтивно зрозумілого набору дій (Рис. 2).

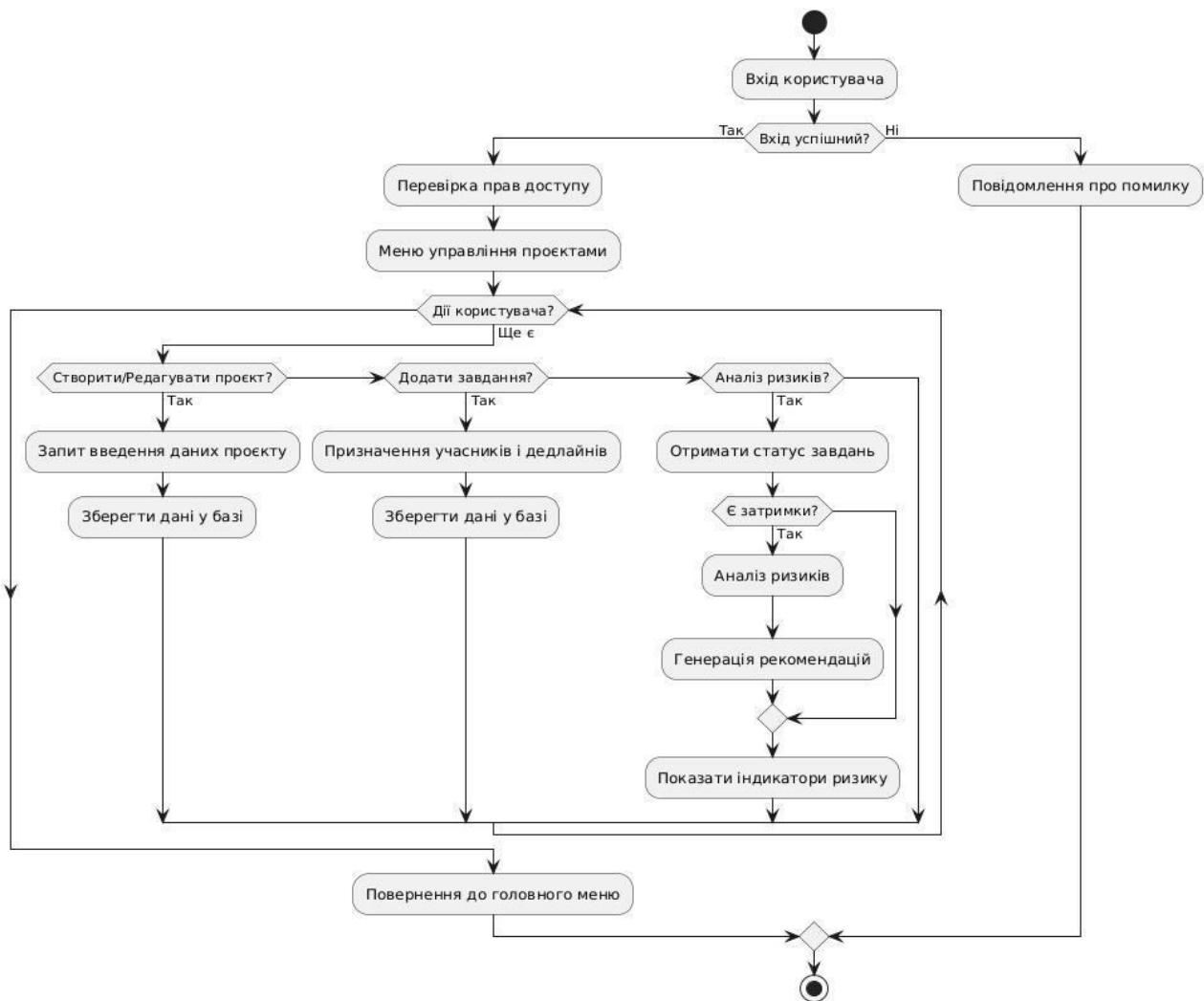


Рис. 2. Алгоритм роботи програми

Для реалізації веб-додатку було створено діаграму компонентів, яка описує логічну структуру системи. Основними елементами є фронтенд, бекенд та база даних.

Кожен із компонентів відповідає за виконання певних функцій, таких як аутентифікація користувачів, обробка даних проєктів та аналіз ризиків (Рис. 3).

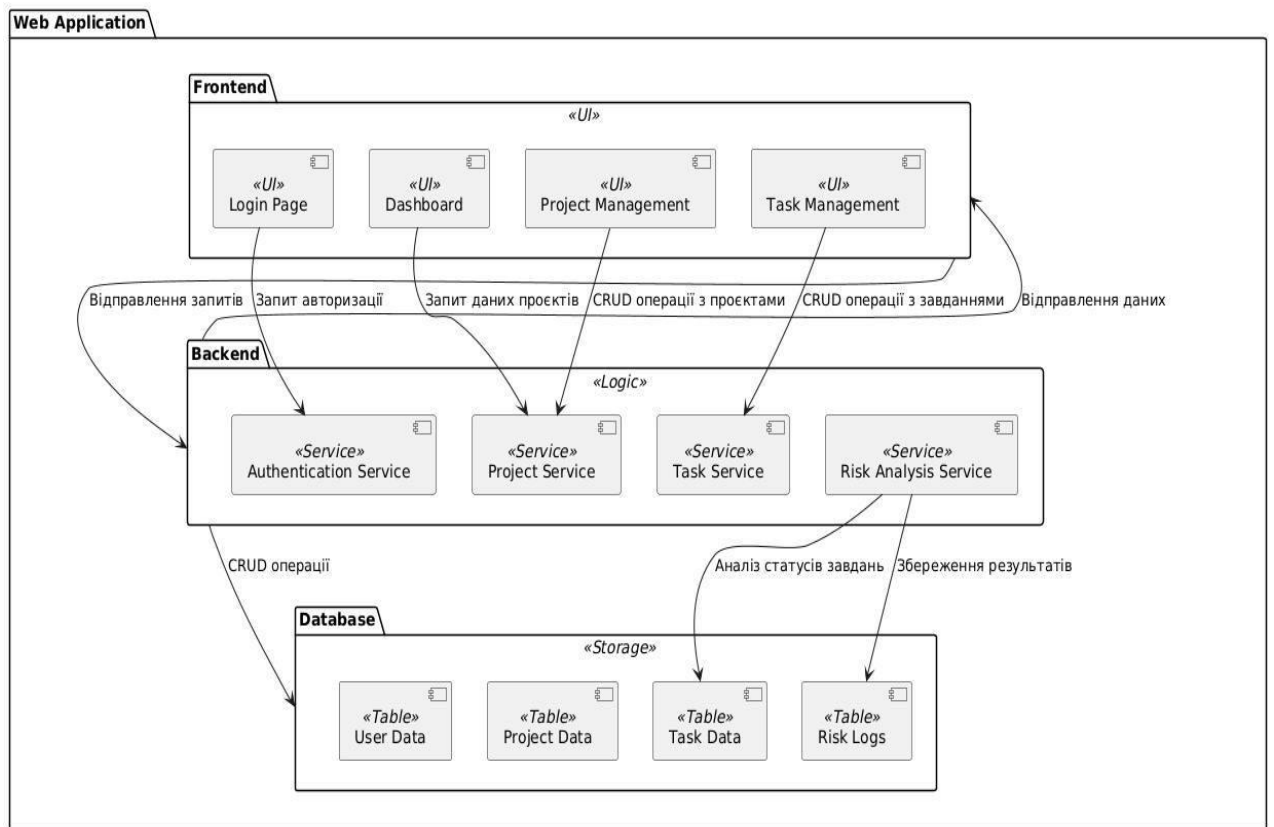


Рис. 3. Діаграма компонентів

Висновки та перспективи.

Запропонований веб-додаток для управління проєктами та відстеження виконання завдань є потужним інструментом для організації роботи команд. Використання Python та JavaScript забезпечує гнучкість і масштабованість, а продумана архітектура дозволяє легко адаптувати систему під потреби користувачів.

Перспективами подальшого розвитку є інтеграція із зовнішніми сервісами для підвищення функціональності додатку, впровадження системи навчання на основі штучного інтелекту для прогнозування ризиків та оптимізації роботи команди.

Список використаних джерел:

1. Розробка веб-додатків на Python | WebCase. *Webcase*. URL: https://webcase.com.ua/uk/blog/razrabotka-veb-prilozhenij-s-ispolzovaniem-python-i-django/?utm_source=chatgpt.com (дата звернення: 25.11.2024).
2. Створення повноцінного додатку з використанням Django та Vue.js. *peerdh.com*. URL: <https://peerdh.com/uk/blogs/programming-insights/crafting-a-full-stack-application-with-django-and-vue-js> (дата звернення: 25.11.2024).
3. Building A CRUD Application With Django Rest Framework And Vue.js - Onlinecode. *onlinecode*. URL: <https://onlinecode.org/building-a-crud-application-with-django-rest-framework-and-vue-js/> (date of access: 25.11.2024).
4. Combining Django and Vue.js: Everything you need to know - Ficus Technologies. *Ficus Technologies*. URL: <https://ficustechnologies.com/blog/combining-django-and-vue-js-everything-you-need-to-know/> (date of access: 25.11.2024).

Використання SSL/TLS для забезпечення конфіденційності і цілісності даних користувача при роботі з зовнішніми ресурсами в мережі

*Василець Данило Богданович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
mag5726325@stud.duikt.edu.ua
Науковий керівник: Соляник Людмила Олексіївна,
к.х.н., доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Постановка задачі.

З розвитком Інтернету та збільшенням кількості сервісів, які надають різноманітні послуги через мережу, питання захисту конфіденційності і цілісності даних стає надзвичайно важливим.

Користувачі постійно передають через Інтернет чутливу інформацію, таку як особисті дані, фінансова інформація, паролі тощо. Ця інформація може бути перехоплена або змінена злочинцями, що призводить до значних збитків як для

користувачів, так і для організацій. Серед ключових загроз з якими може зіштовхнутися користувач будь-якого веб-сервісу варто виділити:

Перехоплення даних (eavesdropping): коли зловмисники можуть перехопити передану інформацію, не змінюючи її, отримуючи таким чином доступ до конфіденційних даних.

Атаки типу "людина посередині" (Man-in-the-Middle): коли зловмисник вставляється між двома комунікуючими сторонами, перехоплює та потенційно змінює інформацію, що передається.

Підробка даних (data tampering): коли зловмисники змінюють передану інформацію, що може призвести до значних негативних наслідків.

Підробка серверів та клієнтів (spoofing): коли зловмисники видають себе за легітимні сервіси або користувачів, щоб отримати доступ до конфіденційних даних або здійснити інші атаки.

Для вирішення цих проблем були розроблені криптографічні протоколи SSL (Secure Sockets Layer) та його наступник TLS (Transport Layer Security), які забезпечують захист даних при їх передачі через мережу. Мета цього дослідження полягає в детальному аналізі механізмів роботи SSL/TLS, їх можливостей та обмежень, а також у визначенні найкращих практик для їх впровадження з метою забезпечення високого рівня конфіденційності та цілісності даних користувачів.

Мета дослідження.

Метою даного дослідження є аналіз використання SSL/TLS протоколів для забезпечення конфіденційності та цілісності даних користувачів при взаємодії з зовнішніми ресурсами в мережі. Зокрема, розглядається ефективність цих протоколів у захисті від атак, що спрямовані на перехоплення та модифікацію даних.

Результати дослідження.

Застосування протоколів SSL/TLS дозволяє значно підвищити рівень безпеки даних, що передаються через Інтернет, завдяки ряду ключових механізмів.

По-перше, протоколи SSL/TLS використовують цифрові сертифікати для перевірки автентичності серверів (і клієнтів у деяких випадках). Це допомагає запобігти атакам типу "людина посередині" та забезпечує впевненість користувача в тому, що він взаємодіє з легітимним сервером.

По-друге, після встановлення з'єднання протоколи SSL/TLS використовують симетричне шифрування для захисту даних, що передаються. Це означає, що дані шифруються перед передачею і розшифровуються тільки отримувачем, що унеможлиблює їх перехоплення та прочитання третіми особами.

По-третє, протоколи SSL/TLS забезпечують цілісність даних за допомогою хеш-функцій та MAC (Message Authentication Code). Це дозволяє виявити будь-які зміни даних під час передачі та запобігає їх підробці.

Нарешті, SSL/TLS використовують асиметричні алгоритми шифрування для безпечного узгодження ключів шифрування. Це забезпечує захищений обмін ключами, які надалі використовуються для симетричного шифрування даних.

В якості основних переваг використання SSL/TLS для захисту даних користувачів було виділено:

Конфіденційність: Усі дані, що передаються між клієнтом і сервером, шифруються, що унеможлиблює їх перехоплення та прочитання третіми особами.

Цілісність: Використання криптографічних хеш-функцій та MAC забезпечує цілісність переданих даних і захищає від їх підробки.

Аутентифікація: Верифікація справжності серверів та клієнтів дозволяє запобігти підробці та забезпечує безпечну комунікацію.

Висновки та перспективи.

Протоколи SSL/TLS є ефективними засобами для забезпечення конфіденційності та цілісності даних при роботі з зовнішніми ресурсами в мережі. Їх впровадження дозволяє значно підвищити рівень безпеки Інтернет-з'єднань, захистити конфіденційні дані користувачів та запобігти різним видам атак.

Результати нашого дослідження показують, що подальший розвиток технологій шифрування, впровадження нових стандартів безпеки, таких як TLS 1.3, а також підвищення обізнаності користувачів про необхідність використання захищених протоколів сприятимуть забезпеченню високого рівня безпеки в Інтернеті у майбутньому.

Список використаних джерел:

1. Dierks T., Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.1. RFC Editor, 2006. <https://doi.org/10.17487/rfc4346>.
2. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3. RFC Editor, 2018. <https://doi.org/10.17487/rfc8446>.
3. "SSL/TLS Strong Encryption: An Introduction". <https://www.sslshopper.com/ssl-tls-strong-encryption.html>.
4. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, 2008. <https://tools.ietf.org/html/rfc5246>.

СЕКЦІЯ 2. БАЗИ ДАНИХ ТА ЗНАНЬ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТА ОБРОБКА ІНФОРМАЦІЇ

Використання Amazon Web Services у логістиці вантажопасажирських перевезень

*Агашков Андрій Юрійович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
Agashkov87@gmail.com
Науковий керівник: Шевченко Світлана Миколаївна,
кандидат педагогічних наук, доцент,
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

В логістиці перевезень вантажу та пасажирів одним із ключових факторів є зв'язок із хмарними технологіями. Хмарні рішення дають змогу автоматизувати ключові логістичні процеси, такі як управління запасами, планування маршрутів та обробка замовлень. Це знижує кількість людських помилок і спрощує управління на кожному етапі ланцюга поставок [1].

На сьогоднішній день існує безліч хмарних технологій, що забезпечують різні потреби та виконують схожі задачі. Використовуючи хмарні платформи на зразок Microsoft Azure або Google Cloud Platform, компанії можуть швидко розробляти індивідуальні рішення для управління ланцюгом поставок, інтегруючи їх з іншими корпоративними системами. Це особливо важливо для великих транспортних компаній, яким необхідно адаптуватися до специфічних вимог різних регіонів або клієнтів, дозволяючи впроваджувати багатoshарові та швидкі логістичні рішення в стислий термін [2, ст.62].

Постановка задачі

Задачею дослідження є аналіз використання Amazon Web Services для логістики вантажопасажирських перевезень.

Мета дослідження.

Визначити критерії вибору Amazon Web Services (AWS) для логістики вантажопасажирських перевезень.

Результати дослідження

Amazon Web Services має наступні ключові переваги над Microsoft Azure, Google Cloud Platform та інші популярні хмарні сервіси [3-4]:

- надає найбільший спектр хмарних сервісів серед усіх провайдерів, пропонуючи понад 200 сервісів для обчислень, зберігання, баз даних, штучного інтелекту, Інтернету речей (IoT) та аналітики. Це дозволяє організаціям використовувати все необхідне в єдиній екосистемі, спрощуючи управління технологічними ресурсами;

- має одну з найрозгалуженіших мереж центрів обробки даних у світі, охоплюючи регіони та зони доступності на всіх континентах. Це дозволяє компаніям розміщувати свої ресурси в безпосередній близькості до кінцевих користувачів, підвищуючи швидкість завантаження та зменшуючи затримки. Конкуренти, такі як Google Cloud і Microsoft Azure, також мають потужну інфраструктуру, але AWS забезпечує більше регіональних варіантів для розгортання додатків;

- забезпечує легку інтеграцію з безліччю сторонніх інструментів і сервісів, що є важливою перевагою для компаній з уже існуючими технологічними рішеннями. Amazon Marketplace також пропонує тисячі продуктів, які можна інтегрувати з AWS, спрощуючи процеси розширення функціональності додатків;

- відповідає багатьом стандартам і сертифікаціям, як-от GDPR, ISO 27001 та HIPAA, що важливо для клієнтів у регульованих галузях, таких як охорона здоров'я та фінанси;

- надає гнучкі моделі ціноутворення, дозволяючи компаніям сплачувати лише за використані ресурси та вибирати між кількома тарифними планами;

Висновки та перспективи

Використання Amazon Web Services у логістиці вантажопасажирських перевезень є обумовленим низкою факторів. Доступ до даних у реальному часі забезпечує прозорість на всіх етапах транспортування, дозволяючи компаніям швидко реагувати на проблеми та зменшувати ризики.

Економія на обладнанні та гнучкість хмарної інфраструктури також допомагають скорочувати витрати, адже компанії платять тільки за використані ресурси. Крім того, хмарні платформи надають потужні інструменти для аналітики та прогнозування, що дозволяє передбачати попит, оптимізувати транспорт і управління запасами.

Список використаних джерел:

1. Диджиталізація в українській логістиці: Роль технологій у оптимізації перевезень. *Landlord*. URL: <https://landlord.ua/news/logistika/dydzhytalizacziya-v-logistyczi-rol-tehnologij-u-optymizacziyi-perevezen-ta-skladuvannya/> (дата звернення: 10.11.2024).
2. Молибога Р. Розвиток цифровізації у транспортно-логістичній галузі з використанням інструментарію платформної концепції // *Успіхи і досягнення у науці*. – 2024. – № 7(7).
3. AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/about-aws/?nc2=h_header (date of access: 10.11.2024).
4. What is Cloud Computing? - Cloud Computing Services, Benefits, and Types - AWS. *Amazon Web Services, Inc.* URL: <https://aws.amazon.com/what-is-cloud-computing/> (date of access: 10.11.2024).

Методика формалізації пошукового запиту на основі методів обробки природньої мови

*Андрюшин Олексій Олексійович,
студент групи ПДМ-52,*

*спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
aleksroebuck@gmail.com*

*Науковий керівник: Худік Богдан Олександрович,
доктор філософії (PhD), ст.викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Обробка природньої мови – загальний напрямок інформатики, штучного інтелекту та математичної лінгвістики, який вивчає проблеми комп'ютерного аналізу та синтезу природньої мови. Прикладні задачі обробки природньої мови охоплюють багато підгалузей, зокрема, виділення інформації, пошук з використанням природньої мови, машинний переклад, ідентифікація плагіату та багато інших.

Постановка задачі

Задачею дослідження є виявлення недоліків в методах обробки природньої мови і їх покращення та виправлення, застосовуючи алгоритм формалізації пошуку.

Мета дослідження

Метою дисертаційної роботи є підвищення ефективності постобробки пошукового запиту, здійсненого природньою мовою, за рахунок використання алгоритмів, які знаходять ключові слова у тексті і формулюватимуть пошуковий запит більш точним.

Результати дослідження

Запропонована методика зможе покращити алгоритми постобробки тексту, здійсненого природньою мовою, за рахунок об'єднання роботи декількох бібліотек постобробки і написання нового алгоритму. Дана методика допоможе проаналізувати існуючі бібліотеки за різними критеріями, зокрема: фоновий шум, якість звуку, чіткість слів тощо. Також, методика включає в себе застосування алгоритму, який

зможе поєднати найкращі якості зазначених бібліотек, зробити пошук ключових слів у тексті і сформулювати більш точний пошуковий запит.

Висновки та перспективи

Результати дослідження мають підкреслити вагомість нових алгоритмів для покращення постобробки природньої мови та формулювання більш чіткого пошукового запиту. Нові методики мають використати в собі найкращі якості вже існуючих методик пост обробки тексту і зробити більш точне формулювання пошукового запиту на основі виділення ключових слів.

Список використаних джерел:

1. Вавіленкова А.І. Структура лінгвістичного процесору системи порівняльного аналізу текстів за змістом /А.І. Вавіленкова // Системи підтримки прийняття рішень. Теорія і практика: зб. доп. наук.-практ. конф. з міжнар. участю.– Київ: ІПММС НАНУ, 2011. – С. 153-156.

Розробка методу автоматичного аналізу великих обсягів текстової інформації за допомогою ASP.NET

Бенедюк Дмитро Володимирович

студент 5 курсу, групи ПДМ-62

Державного університету інформаційно-комунікаційних технологій

benediukdmytro@gmail.com

Науковий керівник: Щербина Ірина Сергіївна,

к.т.н., доц., доцент кафедри Інженерії програмного забезпечення

Державного університету інформаційно-комунікаційних технологій

Постановка задачі

Зі зростанням обсягів текстової інформації, що генерується у цифровому середовищі, виникає необхідність у ефективних методах її автоматичного аналізу. Задача полягає у розробці системи, яка здатна швидко та точно обробляти великі

обсяги тексту, виділяти ключові теми, виявляти тренди та забезпечувати корисні інсайти для користувачів. Основні виклики включають обробку неструктурованих даних, масштабування системи для роботи з великими обсягами інформації та забезпечення високої швидкості обробки [1].

Мета дослідження

Метою даного дослідження є розробка та впровадження методу автоматичного аналізу великих обсягів текстової інформації з використанням платформи ASP.NET [2]. Це включає створення алгоритмів для виділення ключових слів, класифікації тексту, аналізу настроїв та виявлення трендів. Крім того, важливо забезпечити масштабованість та ефективність системи для роботи з великими масивами даних у реальному часі.

Результати дослідження

У ході дослідження було розроблено та впроваджено метод автоматичного аналізу текстової інформації, який включає модулі для виділення ключових слів, класифікації та аналізу настроїв [3]. Використання ASP.NET дозволило створити масштабовану веб-систему, яка інтегрується з різними джерелами даних та забезпечує високу швидкість обробки. Проведені тести показали, що запропонована система успішно справляється з аналізом великих обсягів тексту, забезпечуючи точність та оперативність результатів [4][5].

Висновки та перспективи

Запропонована методика автоматичного аналізу текстової інформації за допомогою ASP.NET продемонструвала свою ефективність та надійність. Система здатна обробляти великі обсяги даних у реальному часі, надаючи користувачам корисні інсайти та допомагаючи у прийнятті рішень. Перспективи подальших досліджень включають вдосконалення алгоритмів аналізу, інтеграцію з додатковими джерелами даних та розширення функціональних можливостей системи для аналізу мультимедійної інформації та соціальних медіа.

Список використаних джерел:

1. Aggarwal, C. C., Zhai, C. Mining Text Data. — Springer, 2012. — 622 с.
2. Manning, C. D., Raghavan, P., Schütze, H. Introduction to Information Retrieval. — Cambridge University Press, 2008. — 482 с.
3. Mehta, P., Parashar, P. Sentiment Analysis and Classification of Text Using Natural Language Processing Techniques and Machine Learning in Social Media Applications // International Journal of Computer Applications. — 2020. — Т. 176, № 36. — С. 1-7.
4. Microsoft ASP.NET Documentation. ASP.NET Core Guide [Електронний ресурс]. — Режим доступу: <https://learn.microsoft.com/en-us/aspnet/core/overview>. — Дата звернення: 24.11.2024.
5. Sebastiani, F. Machine Learning in Automated Text Categorization // ACM Computing Surveys. — 2002. — Т. 34, № 1. — С. 1-47.

Визначення вимог до web-застосунку для пошуку наукових публікацій за ключовими словами

*Василець Данило Богданович,
студент групи ПДМ-53,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
tag5726325@stud.duikt.edu.ua
Науковий керівник: Соляник Людмила Олексіївна,
к.х.н., доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

У сучасному світі кількість наукових публікацій зростає експоненційно. Проте для студентів, викладачів та дослідників все ще існує проблема швидкого доступу до релевантної інформації. Сучасні пошукові системи часто пропонують надмірну кількість нерелевантних результатів. Web-застосунок для пошуку наукових

публікацій за ключовими словами може стати важливим інструментом для оптимізації цього процесу.

Постановка задачі

Розробка web-застосунку, який спрощує процес пошуку наукових публікацій, використовуючи сучасні технології текстового аналізу та API зовнішніх наукових баз даних.

Мета дослідження

Спрощення процесу пошуку наукових публікацій в онлайн-джерелах шляхом створення ефективного та інтуїтивно зрозумілого web-застосунку.

Результати дослідження

Web-застосунок розробляється із застосуванням фреймворку React та інструмента Vite для забезпечення високої продуктивності та зручності розробки.

Основні функціональні можливості застосунку:

1. Пошук за ключовими словами. Автоматичне виправлення орфографії та пропозиції уточнень.
2. Фільтрація результатів. Можливість відбору за роком публікації, авторами та типом документів.
3. Перегляд метаданих. Відображення назви публікації, авторів, року, DOI та короткого опису.
4. Експорт даних. Збереження результатів у форматах CSV або PDF.
5. Інтерактивність. Зручний інтерфейс, що підтримує швидкий пошук завдяки React.

Для інтеграції зі світовими базами наукових даних передбачено використання API, таких як CrossRef, Scopus API та PubMed API. Обробка даних від API здійснюється на стороні серверу через Node.js, що забезпечує оптимізацію запитів і мінімізує затримки.

Архітектура застосунку:

Фронтенд: React з використанням бібліотек Material-UI для створення user-friendly інтерфейсу.

Бекенд: Node.js/Express.js для обробки запитів до API та кешування даних.

Кешування: Redis для пришвидшення повторних запитів.

Висновки та перспективи

Розроблений web-застосунок значно зменшить час на пошук наукових публікацій та забезпечить зручний доступ до релевантної інформації. Це сприятиме покращенню якості навчання та дослідницької діяльності.

У майбутньому передбачається впровадження функцій:

Інтеграція штучного інтелекту для персоналізації результатів пошуку.

Підтримка мультимовного пошуку для роботи з різноманітними базами даних.

Інтеграція з профілями користувачів для збереження історії пошуків та вподобань.

Список використаних джерел:

1. JSON Web Token Introduction - jwt.io – [Електронний ресурс] – Режим доступу : <https://jwt.io/introduction>.
2. Scopus API Documentation – [Електронний ресурс] – Режим доступу: <https://dev.elsevier.com/scopus.html>.
3. CrossRef REST API – [Електронний ресурс] – Режим доступу: <https://www.crossref.org/documentation/retrieve-metadata/rest-api/>.

Адаптивне кешування даних для підвищення ефективності односторінкових веб-додатків

*Власенко Іван Юрійович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
ivanvlasenko206@gmail.com
Науковий керівник: Залива Віталій Вікторович,
доктор філософії (PhD),
ст.викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

У сучасній веб-розробці односторінкові додатки (Single-Page Applications, SPA) набули значної популярності завдяки своїй здатності забезпечувати швидкий та інтерактивний користувацький досвід. Проте, зростання обсягу даних та складності взаємодії з сервером вимагає впровадження ефективних методів управління даними для підтримки високої продуктивності та швидкості завантаження. Одним із ключових підходів до вирішення цієї проблеми є впровадження адаптивних стратегій кешування даних, які дозволяють зменшити навантаження на сервер та покращити відгук додатка.

Постановка задачі

Розробити та впровадити адаптивний механізм кешування даних в односторінкових веб-додатках, який автоматично налаштовується на основі поведінки користувача та характеристик даних, з метою оптимізації продуктивності та зменшення затримок при завантаженні контенту.

Мета дослідження

Підвищити ефективність односторінкових веб-додатків шляхом впровадження адаптивних стратегій кешування даних, що враховують динамічність даних та поведінкові патерни користувачів, для забезпечення швидкого та стабільного доступу до інформації.

Результати дослідження

У ході дослідження було розроблено адаптивний механізм кешування, який включає:

1. Аналіз поведінки користувачів: Відстеження частоти та типів запитів для визначення найбільш затребуваних даних. [1]
2. Класифікація даних: Розподіл даних на статичні та динамічні з урахуванням частоти їх оновлення та важливості для користувача.[2][4]
3. Динамічне налаштування політик кешування: Автоматичне коригування часу життя (TTL) кешу та стратегій оновлення на основі отриманих даних про використання. [3]

Впровадження цього механізму дозволило:

- зменшити кількість запитів до сервера на 40%, що знизило навантаження на серверну інфраструктуру;
- скоротити середній час завантаження сторінки на 30%, покращуючи користувацький досвід;
- підвищити стабільність роботи додатка при високих навантаженнях за рахунок ефективного управління ресурсами.

Висновки та перспективи

Розроблений адаптивний механізм кешування даних демонструє значний потенціал у підвищенні продуктивності односторінкових веб-додатків. Врахування поведінкових патернів користувачів та характеристик даних дозволяє оптимізувати процеси завантаження та відображення контенту. Подальші дослідження можуть бути спрямовані на інтеграцію цього механізму з іншими технологіями оптимізації, такими як передбачувальне завантаження даних та використання машинного навчання для більш точного налаштування політик кешування.

Список використаних джерел:

1. Fowler, M. "Data Fetching Patterns in Single-Page Applications". Addison-Wesley Professional, 2024.

2. Nunes, I. "Adaptive Web Caching: Algorithms and Architectures". Springer, 2020.
- Barker, T. "Intelligent Caching: Advanced Strategies for Web Applications". O'Reilly Media, 2023.
3. Vakali, A., & Giatsoglou, M. "Web Data Management: Concepts and Techniques". Cambridge University Press, 2021.

Огляд особливостей мультиголосового поділу для схожих голосів

*Гапей Максим Юрійович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
math.divergent@gmail.com*

*Науковий керівник: Фесенко Максим Анатолійович,
к.т.н., доц., доцент кафедри Штучного інтелекту
Державного університету інформаційно-комунікаційних технологій*

Помилки у розділенні подібних голосів можуть призвести до сплутування інформації, що є недопустимим в технологіях робототехніки та Інтернету речей. Точний поділ голосів є ключем до створення гнучких та адаптивних систем, однак попри значний прогрес, існуючі методи часто вирішують лише вузький спектр задач, потребуючи оцінки їх доцільності для конкретних умов.

Постановка задачі

Дослідження особливостей мультиголосового поділу для схожих голосів, для створення нейронної мережі, що може стати доповненням до вже існуючих моделей поділу мовлення.

Мета дослідження

Покращення точності мультиголосового поділу для схожих голосів.

Результати дослідження

Основний тон голосу визначає висоту звуку і залежність від вібрації голосових зв'язок. Висота звуку змінюється від напруження зв'язок, їхньої довжини та товщини [1]. Такі особливості дають можливість виділити статеві та вікові характеристики.

Частотний діапазон для чоловіків знаходить в межах 85-225 Гц, жіночий 165-255 Гц, а дитячий близько 250-400 Гц.

Окрім основного тону, людський голос утворює гармоніки, які створюють тембр голосу. Гармоніки можуть досягати частоти у діапазоні до 8-10 кГц. У нижньому діапазоні (до 1 кГц) формується основний тембр, а вищі гармоніки (2-4 кГц) додають ясності голосу, що важливо для розбірливості мови. Також, у будь-якому голосі присутній шум (наприклад, при вимові свистячих та шиплячих приголосних звуків), який знаходиться в діапазоні від 2 кГц до 10 кГц [1]. Варто зазначити, що високочастотний шум підвищує чіткість та впізнаваність звуків, а отже його присутність у мовлення просто необхідна.

Форманти — частоти резонансу, які утворюються під час мовлення (глотка, ротова та носова порожнини) [1]. Вони надають голосу унікальне звучання, і в парі із гармоніками відіграють ключову роль у визначенні індивідуальності голосу. Вважається, що для характеристики людського мовлення достатньо чотирьох формант.

Висновки та перспективи

Спектральні особливості мовця залежать від множини факторів, зокрема статі, віку, гармонік та формант. Ці компоненти мовлення є ключем у визначенні індивідуальності голосу, тому є предметом уваги у вирішенні завдань мультиголосового поділу для схожих голосів [2].

Список використаних джерел

1. Iseli M., Shue Y., Alwan A. Age-And Gender-Dependent Analysis of Voice Source Characteristics. *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Toulouse, 14-19 May 2006. P. 389-392.
2. Prasanna K., Kumaraswamy R. Role of f_0 and formant frequencies in unsupervised separation of convolutive speech mixtures. *Proceedings of the International Conference on Applied and Theoretical Computing and Communication Technology*, Davangere, 29-31 October 2015. P. 316-320.

Методика синтезу шрифту за зразком на основі методів розпізнавання зображень

*Жужков Денис Ігорович
студент групи ПДМ-62*

*спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
ennisonleo@gmail.com;*

*Науковий керівник: Бондарчук Андрій Петрович, д.т.н., професор
Державний університет інформаційно-комунікаційних технологій*

Анотація. Розглянуто питання застосування технологій для розпізнавання шрифтів на основі існуючих в зображеннях текстів

Ключові слова: формування шрифтів; сімейства шрифтів; генерація шрифтів; розпізнавання тексту; розпізнавання шрифтів.

Вступна частин

Розпізнавання тексту дає змогу для використання та обробки великої кількості інформації що допомагає заощаджувати час на перепис або редагування, вже давно існують технології для розпізнавання тексту, а також шрифтів що допомагає зберігати однорідність документу, для можливого відтворення та збереження ідентичності всіх його елементів. Але з кожним часом шрифтів все більше і технології доходять до потреб редагування вже самих зображень або документів. Тому для цього дуже важлива однорідність всього документу чи зображення.

Основна частина

Розпізнавання шрифтів - це процес визначення та ідентифікації шрифту, використаного в тексті, за допомогою спеціалізованих алгоритмів і технологій. Існують різні підходи до цього, але всі вони базуються на аналізі візуальних характеристик символів. Для початку необхідно отримати цифрове зображення це може бути досягнуто завдяки скануванню фізичних документів таких як PDF або jpeg файлу. Це часто використовую для робочих процесів в офісах і бібліотеках де працюють з друкованими книгами, журналами або іншими матеріалами. Або можна

використати фотографування тексту та завантаження з камери смартфона або іншого пристрою. Такий спосіб використовується коли документ є на екрані або фізичний доступ до сканера відсутній. Також текст може вже бути збереженим у цифровому форматі наприклад в зображення або PDF який містить векторне або растрове зображення тексту, що потребує розпізнавання. Для векторного зображення такого який міститься у PDF документах або у форматі SVG в XML документах зберігається у математично визначених об'єктах що дозволяє більш точно передавати інформацію про шрифти. Тобто якщо файл містить векторний текст це значно спрощує процес розпізнавання, оскільки текстові контури легко витягнути і порівняти з базою шрифтів.

Якість зображення має вирішальне значення для успішного пізнавання шрифтів оскільки важливо враховувати роздільну здатність DPI в якій повинна бути не менше 300 точок на дюйм для друкованого тексту оскільки це забезпечує достатню чіткість символів. При низькій роздільній здатності символи можуть бути нечіткими що ускладнює подальший аналіз. Також треба враховувати контрастність оскільки текст має бути чітко Видимо на тлі інакше система може помилково розпізнавати символу або розпізнавати їх некоректно, в найгіршому випадку не побачите зовсім. Тому під час сканування важливо забезпечити належну освітлення та контрастність. Шуми також відіграють важливу роль це можуть бути артефакти на зображення такі як плями, лінії, пил або якесь пошкодження тому чим більше шумів тим складніше виділити символи. Для цього на етапі попередньої обробки застосовуються спеціальні алгоритми для усунення шумів. Також додатково є способи покращення контрастності та різкості зображення для попередньої обробки документу.

Головні проблеми шумів це спотворення таких літер які мають декілька елементів в літері або інші літери які мають якісь додаткові елементи такі як крапки наприклад. Такі елементи характерні для таких латинських літер як “j”, “i” або українських літер «і», «ї» та «й» Що складаються з двох або більше елементів. Такі літери частіше всього є головною проблемою для розпізнавання. Але бувають

моменти коли навіть на чіткому зображенні програмам складно розпізнавати текст оскільки іноді у шрифтах бувають моменти коли самі літери мають дуже маленьку зону або навпаки більшу що дає ефект зліпання літер в одну. Або бувають моменти коли відступ від букв занадто маленький і ці букви можуть бути з литими що додає для сканування складності, і вони відображаються як одна буква.

Також є проблема така як перевантаження зображення тобто коли на зображенні кількість шрифтів є більшою одного таким чином пізнавати конкретний шрифт стає важко. Але найпопулярніші існуючі програми дають змогу вибрати конкретний текст який вони будуть аналізувати. Тобто це буде певна область яку користувач може вибрати сам.

Якщо текст на фотографії був знятий під кутом або не рівно, як це часто трапляється при фотографуванні з телефону, застосовуються алгоритми для вирівнювання тексту та корекції перспективи. Це забезпечує, що літери матимуть правильну форму для подальшого аналізу.

Для розпізнавання шрифтів може бути корисним переведення зображення в чорно-білий режим, оскільки це дозволяє виділити текст і зменшити вплив кольорових артефактів або тіней. Колір важливий лише тоді, коли мова йде про розпізнавання шрифтів, які мають кольорові ознаки.

Після того як зображення було покращено для подальшої обробки та аналізу буде корисним запропонувати користувачу прописувати кожен літеру під зображенням літери яку розпізнала програма. Це забезпечить полегшення пошуку аналогів або створенню конкретних літер на основі виділених із зображення елементів. Також додатково можна запропонувати вписувати текст який був на зображенні що забезпечить полегшення для обробки кожних елементів цього тексту.

Далі система визначає і відокремлює окремі символи або літери на зображенні, які будуть аналізуватися далі. Виділення символів є критичним кроком, оскільки правильна сегментація забезпечує точність розпізнавання шрифту. Основна мета це розділити весь текст на окремі компоненти: символи, слова та рядки. Система

спочатку знаходить область на зображенні, де розташований текст. Це може бути весь документ або лише його частина. Алгоритми можуть знаходити блоки тексту навіть у разі наявності інших елементів, як-от зображень або таблиць. Після виділення текстової області, текст ділиться на рядки. Тут система визначає горизонтальні проміжки між рядками, що дозволяє сегментувати текстовий блок на окремі рядки. Далі відбувається розбиття рядка на окремі слова. Для цього використовується аналіз відстані між символами: система визначає проміжки між літерами всередині слова та між окремими словами. Проміжки між словами значно більші, ніж між літерами всередині слова, що дозволяє чітко розмежувати слова. Після того, як система розділила текст на рядки та слова, відбувається сегментація на рівні окремих символів. Для цього аналізується розмір кожного символу, а також проміжки між літерами.

Використовуються алгоритми які знаходять контури кожного символу на зображенні, що дозволяє відокремити букви від фону. Цей метод використовується для виділення символів на основі вертикальної та горизонтальної проекції зображення тобто аналіз щільності пікселів у кожному рядку та стовпці. Іноді символи можуть бути злиті через погану якість зображення або шрифт. Алгоритми можуть розпізнавати такі випадки й автоматично ділити злиті літери на окремі компоненти. Але додатково треба давати можливість для перевірки.

Після того, як система виділила кожен окремий символ, вона починає аналізувати їх візуальні характеристики. Головною метою буде визначити унікальні особливості символів, що дозволяють ідентифікувати шрифт. Контур символу виділяється і перетворюється в набір векторних даних ліній і кривих. Це допомагає зберегти точну форму кожної букви і порівняти її з контуром символів із бази даних шрифтів. Аналізуються ключові елементи символу, такі як вигини, кути, прямі та криві лінії, що визначають його унікальну форму.

Однією з основних характеристик символів є їх пропорції відношення ширини до висоти. Цей параметр допомагає розрізнити схожі шрифти. Наприклад, символи в шрифтах Arial і Times New Roman мають різні пропорції, хоча можуть здаватися

подібними. Це відношення ширини до висоти кожного символу, яке також використовується для класифікації шрифтів. Алгоритм також вимірює товщину ліній у кожній букві. Це важливий показник, оскільки шрифти можуть мати різні варіації.

Висновки

Процес розпізнавання шрифтів є складним та багатоступеневим. Успішне розпізнавання шрифтів вимагає високоякісного вхідного зображення, потужних алгоритмів сегментації та аналізу символів, а також ефективної бази даних для порівняння характеристик. Що не менш важливе це врахування популярності оскільки на зараз кількість шрифтів с кожним часом все більшає що потребує більш детального аналізу для розпізнавання.

Список використаних джерел

1. Gonzalez, Rafael C., and Richard E. Woods. "Digital Image Processing." методами обробки зображень.
2. Jain, Anil K., and Robert P. W. Duin. "Statistical Pattern Recognition: A Review." Огляд методів розпізнавання образів, у тому числі тексту та символів.
3. Shapiro, Linda G., and George C. Stockman. "Computer Vision." Сегментація, контурне виділення та аналіз об'єктів.
4. Plamondon, Réjean, and Sargur N. Srihari. "Online and Off-line Handwriting Recognition: A Comprehensive Survey." Огляд різних методів розпізнавання тексту та шрифтів.

Покращення продуктивності web-застосунку підтримки клієнтів біржі

*Рудковський Микола Олегович,
студент групи ПДМ-63
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно- телекомунікаційних технологій
rudkovskijkola00@gmail.com
Науковий керівник: Шевченко Світлана Миколаївна,
к.пед.н, доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

У наш час взаємодія між користувачами фінансових бірж і службами підтримки проходить здебільшого онлайн. Особливо актуально це для великих платформ, таких як біржа *Bybit*, де автоматизація підтримки користувачів дозволяє значно підвищити продуктивність і ефективність обслуговування клієнтів. Розробка чат-боту для підтримки користувачів *Bybit* покликана полегшити комунікацію і забезпечити швидку відповідь на найпоширеніші питання, пов'язані з торгівлею, фінансовими операціями та доступом до важливої інформації [1, 2].

Постановка задачі

Задачею дослідження є розробка чат-боту для web-застосунку підтримки біржі *Bybit*, який може стати додатковим засобом допомоги користувачам у вирішенні питань щодо торгівлі та управління рахунками. Чат-бот має забезпечувати доступ до різних функцій: перевірка балансу, інформація про відкриті ордери, актуальні курси валют та інші запити, які часто надходять до служби підтримки.

Мета дослідження

Визначення ключових можливостей та вимог до реалізації і архітектури чат-боту для підтримки біржі *Bybit*. Створення системи, яка дозволяє автоматизовано обробляти типові запити клієнтів, зменшуючи навантаження на операторів служби підтримки, а також надає користувачам зручний інтерфейс для швидкої навігації по основних функціях біржі [3].

Результати дослідження

Для розробки чат-боту на платформі *Yubit* використовується фреймворк *Dialogflow* для обробки природної мови, що дозволяє ефективно розпізнавати запити користувачів і підбирати відповідні відповіді. Як серверний середовище використовується *Node.js*, що забезпечує стабільну і швидку роботу сервісу, а також легку інтеграцію з API *Yubit*. Це рішення дозволяє чат-боту отримувати актуальні дані щодо фінансових операцій, ринкових курсів, стану рахунків тощо.

Чат-бот має архітектуру “клієнт-сервер”. Клієнтом з боку користувача є web-застосунок біржі *Yubit*, через який користувач взаємодіє з ботом. Взаємодія з чат-ботом проходить за допомогою кнопок, текстових команд і передбачених сценаріїв у діалоговому вікні, що дозволяє користувачеві легко орієнтуватися в доступних функціях.

Для роботи з ботом потрібен пристрій з підтримкою браузера або мобільного застосунку *Yubit*, а також підключення до Інтернету. Це робить чат-бот доступним для широкого кола користувачів та знижує залежність від служби підтримки для типових запитів, дозволяючи зосередити ресурси на складніших питаннях.

Висновки та перспективи

Створений продукт має полегшити організацію роботи служби підтримки та надати користувачам швидкий доступ до інформації та основних функцій біржі *Yubit*, що позитивно впливає на їхній користувацький досвід. Обрані технології (*Dialogflow*, *Node.js* та API *Yubit*) надають можливість реалізувати чат-бот з урахуванням усіх вимог, а обрана архітектура забезпечує масштабованість та гнучкість системи.

Крім того, інтеграція чат-боту у web-застосунок *Yubit* створює user-friendly інтерфейс, який не викликає труднощів у використанні більшістю користувачів.

Чат-бот є корисним інструментом, що підвищує продуктивність web-застосунку підтримки біржі *Yubit*, надаючи користувачам можливість швидко та без зайвих зусиль отримувати відповіді на свої запити. Такий підхід знижує навантаження на службу підтримки та дозволяє більш ефективно розподіляти ресурси біржі.

Список використаних джерел:

1. Isinkaye F.O., Folajimi Y.O., Ojokoh B.A. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16, 2015. pp. 261–273. <https://doi.org/10.1016/j.eij.2015.06.005>
2. Misischia, Chiara & Poecze, Flora & Strauss, Christine. (2022). Chatbots in customer service: Their relevance and impact on service quality. *Procedia Computer Science*. 201. 421-428. 10.1016/j.procs.2022.03.055. https://www.researchgate.net/publication/360614232_Chatbots_in_customer_service_Their_relevance_and_impact_on_service_quality
3. Kotsiantis S.B., Zaharakis I., Pintelas P. Supervised machine learning: A review of classification techniques. *Informatica*, 31(3), 2007. pp. 249–268.

Інтеграція потокової та історичної обробки даних для інтелектуального аналізу

*Соломоник Олексій Павлович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
alex.solomonik2000@gmail.com
Науковий керівник: Компан Сергій Володимирович,
к.ф.-м.н. доцент кафедри Інтернет-технологій
Державного університету інформаційно-комунікаційних технологій*

У сучасних інформаційних системах важливим завданням є інтеграція даних із різних джерел для забезпечення високоякісного інтелектуального аналізу. Поєднання потокової обробки даних у реальному часі з історичною аналітикою дозволяє побудувати ефективні рішення, які дають змогу швидко реагувати на зміни та приймати обґрунтовані рішення.

Постановка задачі

Задачею дослідження є розробка архітектури системи, яка дозволяє інтегрувати потокову обробку даних і аналіз історичних даних для застосування в задачах прогнозування, виявлення аномалій та прийняття рішень у реальному часі.

Мета дослідження

Розробити систему, яка об'єднує компоненти потокової обробки даних із сховищами історичних даних (Data Warehouse) для створення аналітичних моделей та забезпечення їхньої ефективності в режимі реального часу.

Результати дослідження

Прототип системи розроблено із використанням наступних інструментів:

1. Apache Kafka для потокової передачі даних у реальному часі.
2. ETL-процеси для завантаження і трансформації даних у сховища.
3. Data Warehouse для зберігання історичних даних.
4. Машинне навчання для прогнозування та виявлення аномалій.

Ключові модулі системи:

1. Джерела даних: сенсори, відео- та аудіопотоки, транзакції.
2. Потокова обробка: модуль реальної обробки (Real-time processing engine) для фільтрації та аналізу даних у режимі реального часу.
3. Аналітика: інтеграція моделей машинного навчання для прогнозування подій та аналізу великих даних.
4. Візуалізація: інформаційні панелі для моніторингу результатів у реальному часі.

Тестування:

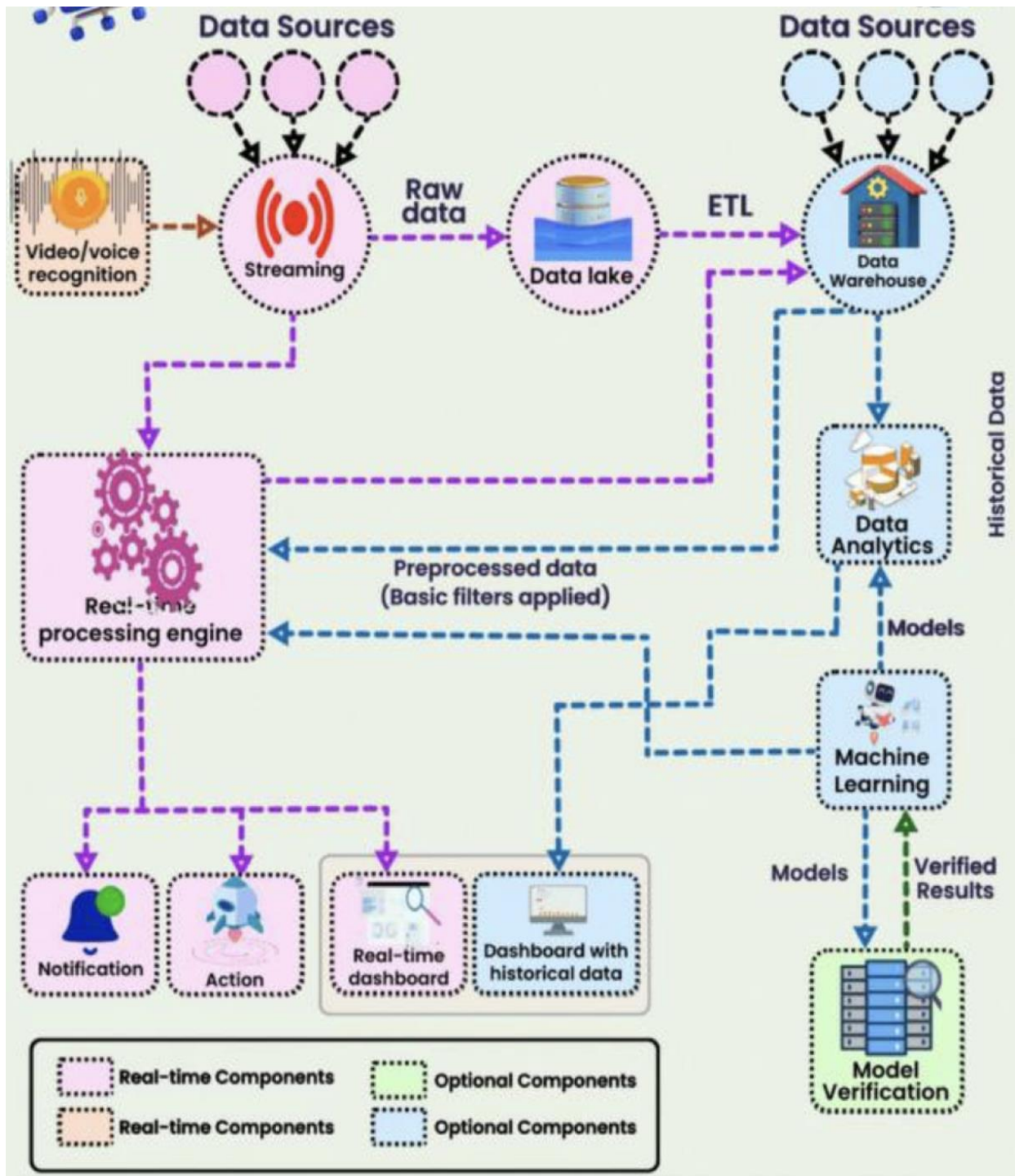


Рис. 1. Архітектура інтеграції поточкових і історичних даних

Система протестована на датасеті транзакцій і показала зменшення часу обробки даних на 40% у порівнянні з традиційними підходами, а точність виявлення аномалій досягла 92%.

Висновки та перспективи

Інтеграція потокової та історичної обробки даних дозволяє побудувати гнучкі системи для інтелектуального аналізу. Впровадження таких рішень допомагає організаціям оперативно реагувати на виклики, знижувати ризики та підвищувати якість прогнозів. Подальший розвиток включає масштабування системи та адаптацію до специфіки різних галузей.

Список використаних джерел:

1. Kumar, A. "Stream Processing and Historical Data Integration for Real-Time Analytics." IEEE Transactions on Big Data, vol. 9, no. 3, pp. 235-246, 2021.
2. Apache Kafka Documentation – [Електронний ресурс]. – Режим доступу: <https://kafka.apache.org>.
3. Chen, D. "Building Data Warehouses for Real-Time Data Integration." Springer, 2021.
4. "ETL Best Practices for Real-Time and Historical Data Processing." TechJournal, vol. 14, pp. 56-69, 2022.
5. "Real-Time Data Analytics with Machine Learning." Proceedings of Big Data Conference, 2022.

Методи та засоби сканування місцевості та візуалізації даних ультразвукового сенсора на основі ехолокації

*Чернявський Ждан Анатолійович
студент 6 курсу, групи ПДМ-61*

*Державного університету інформаційно-комунікаційних технологій
zhdancherniavskiy@gmail.com*

*Науковий керівник: Жебка Вікторія Вікторівна,
доктор технічних наук, професор*

Державного університету інформаційно комунікаційних технологій

Інтернет речей (ІоТ) значно впливає на розвиток промисловості, дозволяючи збирати та обробляти дані з різноманітних сенсорів для моніторингу й оптимізації виробничих процесів. Однією з перспективних технологій у сфері ІоТ є ультразвукові сенсори, які використовуються для сканування, контролю стану обладнання та виявлення перешкод.

Постановка задачі

Основна задача полягає у створенні методу для збору й візуалізації даних з ультразвукових сенсорів, який забезпечить точне і надійне сканування промислових об'єктів. Це дозволить оптимізувати використання ресурсів, покращити процеси контролю та забезпечити безпеку на виробництві.

Мета дослідження

Метою дослідження є розробка методології, яка інтегрує ультразвукове сканування з ІоТ-платформами для створення 3D моделей промислових об'єктів. Таке рішення сприятиме ефективному збору й обробці даних, а також підвищенню продуктивності виробничих процесів.

Результати дослідження

Розроблена ІоТ-система на основі ультразвукових сенсорів дозволяє сканувати простір та забезпечувати його візуалізацію. Проведені експерименти продемонстрували точність до 5 см, що є достатнім для більшості промислових

завдань. Система дозволяє оперативно виявляти потенційні проблеми та підвищувати безпеку в умовах виробництва.

Висновки та перспективи

Запропонований метод є ефективним інструментом для створення 3D моделей в реальному часі, що робить його перспективним для застосування у сфері робототехніки та автоматизації процесів. Подальші дослідження можуть бути спрямовані на покращення точності сенсорів на великих відстанях та інтеграцію алгоритмів штучного інтелекту для автоматизованого аналізу отриманих даних.

Список використаних джерел:

1. What Is an Ultrasonic Sensor?. URL: <https://maxbotix.com/blogs/blog/how-ultrasonic-sensors-work>
2. Helicopter wire strike protection and prevention devices: Review, challenges, and recommendations URL: <https://www.sciencedirect.com/topics/engineering/ultrasonic-sensor>
3. Ultrasonic Sensors: How They Work (and How to Use Them with Arduino). URL: <https://www.arrow.com/en/research-and-events/articles/ultrasonic-sensors-how-they-work-and-how-to-use-them-with-arduino>

Порівняльний аналіз систем предиктивної аналітики

*Яценко Дмитро Володимирович,
аспірант групи АПЗ-11,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
dmitry.yatsyk@gmail.com
Науковий керівник: Садовенко Володимир Сергійович,
кандидат фізико-математичних наук, доцент,
професор кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Предиктивна аналітика є важливим інструментом для прийняття управлінських рішень у сфері маркетингу. Вона дозволяє прискорити процес прогнозування значень показників та майбутніх подій. Різноманіття доступних систем предиктивної аналітики ускладнює вибір оптимального рішення для конкретних завдань.

Постановка задачі.

Задачею дослідження є порівняння основних систем предиктивної аналітики, визначення їх сильних та слабких сторін, а також визначення їх сценаріїв використання.

Мета дослідження.

Визначення ключових можливостей основних систем предиктивної аналітики.

Результати дослідження.

У дослідженні проаналізовано такі системи, як: AutoML [1], Vertex AI [2], IBM Watson Studio [3], SAS Viya [4], Altair RapidMiner [5], KNIME Analytics Platform [6], Dataiku [7], Azure Machine Learning [8], Amazon QuickSight [9], Amazon SageMaker [10], Oracle Analytics Cloud [11], Adobe Analytics [12], Alteryx AI Platform for Enterprise Analytics [13], Databricks [14], SAP Analytics Cloud [15], ChannelMix [16], H2O Driverless AI [17], IBM SPSS [18], Spotfire [19], Qlik [20].

Назва	Опис	Переваги	Недоліки	Сценарії використання
AutoML	Автоматизує процес побудови моделей машинного навчання, зменшуючи потребу в ручному налаштуванні.	Швидке прототипування, зручність для користувачів без глибоких знань ML.	Менший контроль над моделями, обмеженість у налаштуваннях.	Початкові дослідження, швидкий аналіз даних.
Vertex AI	Платформа від Google Cloud для розробки та розгортання моделей ML.	Інтеграція з Google Cloud, підтримка AutoML та кастомних моделей.	Вимагає досвіду з Google Cloud, вартість може зрости при масштабуванні.	Великі проекти з потребою в масштабуванні та інтеграції з іншими сервісами Google.
IBM Watson Studio	Платформа для даних та AI з інструментами для співпраці.	Підтримка різних мов програмування, потужні інструменти для аналізу. Інтуїтивний інтерфейс та широкий спектр статистичних методів.	Складність для новачків, високі вимоги до ресурсів.	Корпоративні проекти з великою командою аналітиків. Підходить для користувачів, які потребують глибокого статистичного аналізу без глибоких знань програмування.
SAS	Хмарна	Потужні	Висока	Великі

Viya	платформа для аналітики та штучного інтелекту.	статистичні інструменти, масштабованість.	вартість, крута крива навчання.	підприємства з потребою в глибокому аналізі даних.
Altair RapidMiner	Платформа для підготовки даних та машинного навчання без коду.	Інтуїтивний інтерфейс, швидке створення моделей.	Обмежена гнучкість, може не підійти для складних задач.	Малий та середній бізнес, швидкий аналіз даних.
KNIME Analytics Platform	Відкрита платформа для аналітики з модульним підходом.	Безкоштовна, активна спільнота, розширюваність.	Може вимагати додаткових плагінів, обмежена підтримка.	Академічні дослідження, прототипування.
Dataiku	Платформа для спільної роботи над даними та AI проектами.	Колаборація, підтримка повного циклу даних.	Вартість для великих команд, складність налаштування.	Командні проекти, корпоративна аналітика.
Azure Machine Learning	Сервіс від Microsoft для створення та розгортання моделей ML.	Інтеграція з Azure, потужні інструменти для DevOps.	Вимагає знань Azure, може бути дорого для масштабних задач.	Проекти в екосистемі Microsoft, масштабовані рішення.
Amazon QuickSight	Сервіс BI для візуалізації та аналізу даних.	Швидка візуалізація, інтеграція з AWS.	Обмежені можливості ML, більше фокусується на BI.	Створення дашбордів, базовий аналіз.

Amazon SageMaker	Повна платформа для розробки, навчання та розгортання ML моделей.	Масштабованість, інтеграція з AWS сервісами.	Вимагає досвіду з AWS, вартість може зрости.	Розробка кастомних моделей, великі ML проекти.
Oracle Analytics Cloud	Хмарна платформа для аналітики та прогнозування.	Інтеграція з Oracle DB, потужні аналітичні інструменти.	Висока вартість, складність інтеграції з іншими системами.	Підприємства, що використовують Oracle екосистему.
Adobe Analytics	Інструмент для веб-аналітики та поведінкового аналізу.	Глибокий аналіз користувацької поведінки, інтеграція з Adobe Marketing Cloud.	Фокус на веб-аналітиці, висока вартість.	Маркетингові кампанії, аналіз веб-трафіку.
Alteryx AI Platform for Enterprise Analytics	Платформа для підготовки даних та аналітики без коду.	Швидкість, інтеграція з різними джерелами даних.	Обмежена гнучкість, вартість ліцензій.	Бізнес-аналітика, швидке прийняття рішень.
Databricks	Платформа для обробки великих даних та ML на базі Apache Spark.	Масштабованість, підтримка кількох мов програмування.	Складність налаштування, вартість при великих обсягах даних.	Великі дані, складні ML задачі.
SAP Analytics	Платформа для аналітики, планування та	Інтеграція з SAP продуктами,	Висока вартість, обмеження	Підприємства, що використовують

Cloud	прогнозування.	комплексність.	поза SAP екосистемою.	ть SAP рішення.
Channel Mix	Платформа для маркетингової аналітики та консолідації даних.	Об'єднання даних з різних каналів, спеціалізація на маркетингу.	Обмежена універсальність, специфічність у галузі.	Маркетингові агенції, аналіз кампаній.
H2O Driverless AI	Автоматизована платформа для машинного навчання.	Швидкість моделювання, пояснюваність моделей.	Вартість ліцензії, вимагає потужних ресурсів.	Швидке створення моделей, проекти з вимогами до пояснюваності.
IBM SPSS	Статистичне програмне забезпечення для аналізу даних.	Широкий набір статистичних методів, надійність.	Застарілий інтерфейс, обмежені можливості ML.	Академічні дослідження, соціальні науки.
Spotfire	Платформа для візуальної аналітики та прогнозування.	Інтерактивні дашборди, швидка візуалізація.	Вартість, складність для новачків.	Бізнес-аналітика, презентації даних.
Qlik	Інструмент для самостійної бізнес-аналітики та візуалізації.	Асоціативний пошук, швидкий доступ до інсайтів.	Крута крива навчання, додаткові витрати на підтримку.	Аналіз продажів, управлінські звіти.

Висновки та перспективи.

Вибір системи предиктивної аналітики залежить від специфічних потреб, масштабу проекту, наявних ресурсів та галузевої специфіки. Важливо враховувати як технічні можливості платформи, так і її відповідність бізнес-цілям організації. Проведений аналіз має полегшити підприємствам процес вибору системи предиктивної аналітики.

Список використаних джерел:

1. AutoML, <https://cloud.google.com/automl>
2. Vertex AI, <https://cloud.google.com/vertex-ai>
3. IBM Watson Studio, <https://www.ibm.com/products/watson-studio>
4. SAS Viya, https://www.sas.com/en_us/software/viya.html
5. Altair RapidMiner, <https://altair.com/altair-rapidminer>
6. KNIME Analytics Platform, <https://www.knime.com/>
7. Dataiku, <https://www.dataiku.com/>
8. Azure Machine Learning, <https://azure.microsoft.com/en-us/products/machine-learning/>
9. Amazon QuickSight, <https://aws.amazon.com/ru/quicksight/>
10. Amazon SageMaker, <https://aws.amazon.com/ru/sagemaker/>
11. Oracle Analytics Cloud, <https://www.oracle.com/in/business-analytics/analytics-platform/>
12. Adobe Analytics, <https://business.adobe.com/products/analytics/adobe-analytics.html>
13. Alteryx AI Platform for Enterprise Analytics, <https://www.alteryx.com/>
14. Databricks, <https://www.databricks.com/>
15. SAP Analytics Cloud, <https://www.sap.com/products/technology-platform/cloud-analytics.html>
16. ChannelMix, <https://channelmix.com/>
17. H2O Driverless AI, <https://h2o.ai/platform/ai-cloud/make/h2o-driverless-ai/>
18. IBM SPSS, <https://www.ibm.com/spss>
19. Spotfire, <https://www.spotfire.com/overview>
20. Qlik, <https://www.qlik.com/us>

Визначення вимог до паспортизації діджитал продуктів на основі блокчейну

*Гашко Андрій Олександрович,
аспірант 2го курсу, Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
a.gashko1111@gmail.com
Науковий керівник: Бондарчук Андрій Петрович,
д.т.н., професор,
Державного університету інформаційно комунікаційних технологій.*

Постановка задачі.

Задачею дослідження є розробка декількох web3 додатків та досягнення консорціуму між ними за допомогою смарт контракту завантаженого у блокчейн для більш яснішого розуміння вимог що потребує паспортизація діджитал продуктів.

Мета дослідження.

Обміркувати останні успішні кейси запровадження смарт контрактів у соціальні процеси на базі блокчейну та майбутнє технології в Європі.

Результати дослідження.

Результатом дослідження є створення AML моделі даних для створення консорціуму між двома web3 додатками за умови відсутності єдиноособного контролю. Це дуже важливо для побудови здорової екосистеми web3 додатків які будуть займатись, наприклад збором та зберіганням соціальних балів, що до прав людини або її продуктивності.

Однією із перешкод у реалізації подібних проектів є те що довіра такої конфіденційної інформації одній централізованій організації може назавжди скомпроментувати довіру до галузі смарт контрактів і стати причиною для перешкод для компаній впроваджувати блокчейн у свої бізнес процеси та продовжувати перебудову своїх бізнес моделей та ERP систем.

Blockchain забезпечує децентралізацію, гарантуючи, що жодна особа не має доступу до всієї інформації. Натомість компанії керують своїм блокчейн

вузлом(ною) і за умовами смарт контракту можуть зберігати свою інформацію локально, без її спільного використання чи доступу комусь іншому.

Упови смарт контракту забезпечують правила для до AML моделі для подальшої роботи та обчислень цієї конфіденційної інформації та дозволяє ділитись лише необхідними результатами цих розрахунків, не передаючи основних корпоративних даних. Цей підхід гарантує захист конфіденційної інформації, водночас дозволяючи забезпечити необхідну прозорість у співпраці та трудових або логістичних ланцюгах.

Висновки та перспективи.

Не зважаючи на те що блокчейн забезпечує міцну основу для DPP, його розгортання стикається з такими проблемами, як масштабованість, споживання енергії та складність створення децентралізованої інфраструктури. Однак, потенціал для інновацій залишається величезним, включаючи використання смарт-контрактів для просування соціальних та екологічних практик а також вирішення проблем узгодження прозорості діяльності та конфіденційності.

Список використаних джерел:

1. Jiayi Sun, Wensheng Gan, Han-Chieh Chao, Philip S Yu, and Weiping Ding. 2023. Internet of behaviors: A survey. IEEE Internet of Things Journal. DOI: 10.1109/IJOT.2023.3247594 (2023), 1–18.
2. Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. 2018. Blockchain technology overview. Technical Report. National Institute of Standards and Technology, U.S. Department of Commerce. <https://www.nist.gov/publications/blockchaintechnology-overview>
3. Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. IEEE Journal on Selected Areas in Communications 37, 6 (2019), 1205–1221.
4. 2021 Global Crypto Adoption Index: Worldwide Adoption Jumps Over 880% With P2P Platforms Driving Cryptocurrency Usage in Emerging Markets, <https://blog.chainalysis.com/reports/2021-global-crypto-adoption-index/>

Порівняльний аналіз продуктивності ORM-технологій для типових CRUD-операцій у веб-застосунках на основі C# та ASP.NET

Побережник Андрій Анатолійович

Студент 6 курсу, групи ПДМ-62

Державного університету телекомунікацій

andrewdev2021@gmail.com

Науковий керівник: Соляник Людмила Олесіївна,

к.х.н., доцент кафедри Інженерії програмного забезпечення

Державного університету інформаційно-комунікаційних технологій

Постановка задачі

В наш час веб-застосунки на основі C# та ASP.NET широко використовуються для створення сучасних додатків, де ефективність взаємодії з базами даних є ключовим чинником. ORM-технології, такі як Entity Framework, Dapper та інші, надають можливість працювати з базами даних за допомогою об'єктно-орієнтованого підходу. Однак вибір ORM значно впливає на продуктивність веб-застосунків, особливо у випадках типових CRUD-операцій (створення, читання, оновлення та видалення даних). Існує потреба у систематичному порівнянні продуктивності популярних ORM-рішень для визначення оптимальної технології в залежності від вимог застосунку, таких як швидкодія, використання ресурсів і простота реалізації [1].

Мета дослідження

Метою дослідження є проведення порівняльного аналізу продуктивності популярних ORM-технологій у контексті виконання CRUD-операцій у веб-застосунках, створених на C# та ASP.NET. Дослідження включає оцінку Entity Framework, Dapper та інших ORM-рішень за такими критеріями, як швидкість виконання запитів, споживання ресурсів, складність реалізації та зручність інтеграції з платформою ASP.NET. Результати спрямовані на виявлення найефективнішого ORM-рішення для типових задач веб-застосунків [3].

Результати дослідження

Продуктивність ORM-технологій суттєво залежить від типу виконуваних операцій і розміру оброблюваних даних. Entity Framework забезпечує високу зручність завдяки вбудованим інструментам і високому рівню абстракції, але іноді демонструє нижчу швидкість порівняно з Dapper, особливо при виконанні операцій читання великих обсягів даних. Dapper, навпаки, забезпечує вищу продуктивність завдяки мінімальному рівню абстракції, але вимагає більшої кількості коду для реалізації складних запитів. Результати дослідження підтверджують, що вибір ORM-технології має базуватися на специфіці веб-застосунку, обсягах даних та типах операцій, які виконуються найчастіше [4],[5].

Висновки та перспективи

Запропонований порівняльний аналіз ORM-технологій дозволяє розробникам веб-застосунків на C# та ASP.NET більш чітко обирати ORM-рішення залежно від потреб конкретного проекту. Entity Framework підходить для невеликих проектів або систем із типовими запитамі, тоді як Dapper оптимальний для високонавантажених систем. У подальшому дослідження можуть бути зосереджені на оцінці продуктивності ORM у сценаріях високих навантажень, інтеграції з хмарними сервісами та роботи з розподіленими базами даних [2].

Список використаних джерел

1. Albahari, J., & Albahari, B. "C# 10 and .NET 6 in a Nutshell." O'Reilly Media, 2022.
2. Microsoft Docs. "Entity Framework Core Documentation." [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/ef/core/>.
3. Marc Gravell. "Dapper.NET: High-performance Micro-ORM for .NET." [Електронний ресурс] – Режим доступу: <https://dapper-tutorial.net/>.
4. Kuhn, T. "Performance Optimization of Database Applications in ASP.NET." Springer, 2020.
5. Fowler, M. "Patterns of Enterprise Application Architecture." Addison-Wesley, 2002.

Розробка методу лексичної валідації та корекції слів на основі словника у текстах українською мовою

*Шульга Сергій Сергійович,
студент 5 курсу, групи ПДМ-52
Державного університету інформуваційно-комунікаційних технологій
sergey2002shulga@gmail.com
Науковий керівник: Худік Богдан Олександрович,
доктор філософії (PhD),
ст. викладач кафедри Інженерії програмного забезпечення
Державного університету інформуваційно-комунікаційних технологій*

Постановка задачі

В умовах швидкого розвитку інформаційних технологій та зростання використання цифрових пристроїв виникає потреба у забезпеченні коректності текстів, що вводяться користувачами. Часто зустрічаються ситуації, коли користувач випадково вводить текст не тією мовою через неправильно налаштовану розкладку клавіатури. Це призводить до необхідності корекції текстів, що ускладнює роботу користувача, знижує ефективність комунікації та загальну швидкість взаємодії. Особливо актуальною ця проблема стає в умовах мультимовного середовища, де користувачі регулярно змінюють мову введення під час роботи з текстами. Неправильно введені слова можуть призводити до непорозумінь, помилок у важливих документах та додаткових витрат часу на виправлення помилок.

Таким чином, актуальною є задача створення системи, яка б автоматично виявляла та виправляла неправильно введені слова, тим самим покращуючи ефективність роботи користувачів та якість створюваних текстів.

Мета дослідження

Метою даного дослідження є розробка методу автоматичної корекції слів у текстах українською мовою, що вводяться користувачем. Метод повинен працювати в фоновому режимі та автоматично визначати неправильні слова, замінюючи їх на правильні відповідно до мови введення, та, при необхідності, змінювати мову введення на правильну. Основою методу буде використання перевірки слів у

словниках відповідних мов. Це забезпечить високу точність і швидкість корекції тексту, оскільки перевірка буде базуватися на заздалегідь визначених словниках, що містять правильні варіанти написання слів для кожної мови.

Результати дослідження

У ході дослідження було розроблено алгоритм, який використовує метод перевірки слів у словниках відповідних мов для аналізу введеного тексту та його корекції. Алгоритм включає такі етапи:

1. Виявлення неправильно введених слів за допомогою порівняння з базою даних словників для різних мов.

2. Автоматичне визначення мови введення на основі аналізу структури та частотності символів у введеному тексті.

3. Заміна неправильно введених слів на правильні варіанти з використанням словника, а також автоматична зміна мови введення, якщо це необхідно.

Тестування розробленого методу показало високу точність виявлення та корекції неправильно введених слів, що дозволяє значно підвищити зручність використання текстових редакторів та інших програм, що потребують введення тексту.

Алгоритм продемонстрував високу швидкість роботи та незначне навантаження на системні ресурси, що робить його придатним для використання на широкому спектрі пристроїв.

Висновки та перспективи

Розроблений метод корекції слів у текстах українською мовою є ефективним інструментом для покращення точності введення тексту. Він дозволяє автоматично виправляти неправильно введені слова, що значно знижує ймовірність помилок та підвищує ефективність роботи користувачів. В подальшому планується розширення функціональності алгоритму для підтримки інших мов та інтеграції з популярними текстовими редакторами та месенджерами. Це дозволить забезпечити більш широку аудиторію користувачів зручним інструментом для автоматичної корекції текстів та підвищення продуктивності їхньої роботи. Додатково, розглядається можливість

впровадження алгоритму в мобільні додатки та веб-сервіси, що розширить його застосування та підвищить зручність використання в різних умовах.

Список використаних джерел:

1. Ковальчук А.П. Словники та системи корекції текстів. – Київ: Видавництво Київського університету, 2018. – 256 с.
2. Гавриленко М.І. Технології обробки текстової інформації. – Львів: Палітра, 2019. – 312 с.
3. Доленко О.В. Автоматизація процесів введення тексту. – Харків: Фоліо, 2020. – 290 с.

Алгоритм аналізу слів за допомогою словника в системі корекції текстів українською мовою

*Шульга Сергій Сергійович,
студент 5 курсу, групи ПДМ-52
Державного університету інформукаційно-комунікаційних технологій
sergey2002shulga@gmail.com
Науковий керівник: Худік Богдан Олександрович,
доктор філософії (PhD),
ст. викладач кафедри Інженерії програмного забезпечення
Державного університету інформукаційно-комунікаційних технологій*

Постановка задачі.

Одним із ключових етапів розробки системи автоматичної корекції текстів є створення ефективного алгоритму аналізу слів за допомогою словника. З огляду на зростаючу потребу в забезпеченні коректності текстів, які вводяться користувачами, актуальним є завдання розробки автономного методу перевірки слів. Такий метод повинен забезпечувати високу швидкість і точність аналізу, що є особливо важливим в умовах роботи з великими обсягами текстових даних. Необхідність автономної

роботи також продиктована можливими обмеженнями в доступі до Інтернету та залежністю від зовнішніх сервісів, які можуть бути недоступними або працювати з перебоями.

Мета дослідження.

Метою даного дослідження є розробка алгоритму аналізу слів за допомогою словника, що буде використовуватися в системі автоматичної корекції текстів українською мовою. Алгоритм повинен працювати автономно, без необхідності підключення до Інтернету, та забезпечувати високу швидкість і точність перевірки слів. Основною задачею алгоритму є виявлення неправильно введених слів шляхом їх перевірки у відповідних словниках.

Результати дослідження.

У ході дослідження було розроблено алгоритм, що включає такі етапи:

1. Аналіз введеного тексту та розбиття його на окремі слова.
2. Перевірка кожного слова на наявність у словнику відповідної мови. Якщо слово є у словнику, воно вважається правильним. Якщо слова немає у словнику, воно вважається написаним неправильно і потребує заміни.
3. Заміна неправильно введених слів на правильні варіанти з використанням словника.

Основними перевагами даного методу є:

1. Автономність. Алгоритм не залежить від наявності Інтернету, оскільки всі словники зберігаються у базі даних на локальному сервері. Це забезпечує стабільну роботу системи в будь-яких умовах.
2. Швидкість. Завдяки прямій роботі зі словником, швидкість пошуку є високою і може бути наближеною до лінійної. Це дозволяє обробляти великі обсяги тексту за короткий час.
3. Незалежність від зовнішніх сервісів. Використання локальних словників виключає залежність від зовнішніх API, які можуть бути платними, недоступними або

працювати з перебоями. Це також підвищує безпеку даних користувача, оскільки вся інформація обробляється локально.

Тестування алгоритму показало високу ефективність і точність виявлення та корекції неправильно введених слів. Алгоритм продемонстрував стабільну роботу та високу швидкість обробки тексту навіть на великих обсягах даних.

Висновки та перспективи.

Розроблений алгоритм аналізу слів за допомогою словника є ефективним інструментом для покращення точності введення тексту. Він дозволяє автоматично виявляти та виправляти неправильно введені слова, що значно знижує ймовірність помилок та підвищує ефективність роботи користувачів.

В подальшому планується розширення функціональності алгоритму для підтримки інших мов та інтеграції з популярними текстовими редакторами та месенджерами.

Це дозволить забезпечити більш широку аудиторію користувачів зручним інструментом для автоматичної корекції текстів та підвищення продуктивності їхньої роботи.

Список використаних джерел:

1. Ковальчук А.П. Словники та системи корекції текстів. – Київ: Видавництво Київського університету, 2018. – 256 с.
2. Тарасенко В.В. Обробка природньомовних текстів. – Одеса: Чорномор'я, 2017. – 278 с.
3. Мельник П.С. Сучасні технології корекції текстів. – Дніпро: Ліра, 2021. – 304 с.

Архітектура системи автоматичного аналізу текстової інформації на базі ASP.NET

*Бенедюк Дмитро Володимирович
студент 5 курсу, групи ПДМ-62
Державного університету телекомунікацій
+380961224545*

benediukdmytro@gmail.com

*Науковий керівник: Щербина Ірина Сергіївна,
Кандидат технічних наук, доцент,
Доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Постановка задачі

В еру інформаційних технологій щодня генеруються величезні обсяги текстової інформації, аналіз якої може надати цінні дані для прийняття рішень. Проте традиційні методи обробки даних не завжди здатні забезпечити швидкість та масштабованість, необхідні для роботи з такими великими масивами тексту[1]. Задача полягає у створенні архітектури системи, яка, використовуючи можливості платформи ASP.NET, здатна автоматично обробляти великі обсяги текстової інформації в режимі реального часу, виділяти ключові теми, здійснювати класифікацію та аналіз настроїв, а також виявляти тренди.

Мета дослідження

Основною метою дослідження є розробка архітектури системи автоматичного аналізу текстової інформації на базі ASP.NET, що дозволяє досягти високої продуктивності та масштабованості для аналізу текстових даних[2]. Система має забезпечувати інтеграцію з різними джерелами даних, підтримку алгоритмів обробки тексту для виділення ключових слів, класифікації, аналізу настроїв і виявлення трендів, а також надавати можливість роботи в реальному часі для отримання швидких і точних результатів.

Результати дослідження:

У процесі дослідження було розроблено архітектуру системи автоматичного аналізу текстової інформації на базі ASP.NET, яка забезпечує високу продуктивність при обробці великих обсягів даних. Використання асинхронного

програмування та кешування дозволило досягти оптимальної швидкості обробки. Система інтегрується з різними джерелами даних, такими як API, бази даних та веб-ресурси, що дозволяє безперервно збирати текстові дані для подальшого аналізу[3][4]. Результати тестування продемонстрували, що запропонована архітектура ефективно обробляє значні обсяги тексту, забезпечуючи користувачів точними та актуальними результатами.

Висновки та перспективи:

Розроблена архітектура автоматичного аналізу текстової інформації на базі ASP.NET показала високу ефективність при обробці великих обсягів текстових даних у реальному часі. Система є легкою в інтеграції з різними джерелами даних, що робить її універсальною для широкого спектра застосувань. Подальші дослідження можуть бути спрямовані на вдосконалення алгоритмів аналізу тексту, інтеграцію додаткових джерел даних, а також розширення можливостей системи для роботи з мультимедійною інформацією та аналізу соціальних медіа[5].

Список використаних джерел:

1. Aggarwal, C. C., Zhai, C. Mining Text Data. — Springer, 2012. — 622 с.
2. Manning, C. D., Raghavan, P., Schütze, H. Introduction to Information Retrieval. — Cambridge University Press, 2008. — 482 с.
3. Mehta, P., Parashar, P. Sentiment Analysis and Classification of Text Using Natural Language Processing Techniques and Machine Learning in Social Media Applications // International Journal of Computer Applications. — 2020. — Т. 176, № 36. — С. 1-7.
4. Microsoft ASP.NET Documentation. ASP.NET Core Guide [Електронний ресурс]. — Режим доступу: <https://learn.microsoft.com/en-us/aspnet/core/overview>. — Дата звернення: 24.11.2024.
5. Sebastiani, F. Machine Learning in Automated Text Categorization // ACM Computing Surveys. — 2002. — Т. 34, № 1. — С. 1-47.

СЕКЦІЯ 3. МЕТОДИ ТЕСТУВАННЯ, ВЕРИФІКАЦІЇ, ОЦІНЮВАННЯ ЯКОСТІ ТА ОПТИМІЗАЦІЇ ПРОГРАМ

Аналіз недоліків автоматизованого тестування UX/UI дизайну на основі методів штучного інтелекту

*Бажан Юрій Павлович,
аспірант групи АІПЗ№-11,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
iurii.bazhan@gmail.com
Науковий керівник: Золотухіна Оксана Анатоліївна,
кандидат технічних наук, доцент,
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

UX/UI дизайн є важливим етапом розробки програмного забезпечення, оскільки визначає, як користувачі взаємодіють з застосунками. Відповідно, тестування UX/UI дизайн програмних продуктів має бути ретельним та ефективним. Ручне та автоматизоване тестування часто не можуть повністю охопити складність користувацького досвіду, тому автоматизація тестування UX/UI з використанням методів штучного інтелекту є перспективним напрямом [1]. Однак існують певні недоліки, які можуть виникнути при застосуванні ШІ.

Постановка задачі.

Дослідити недоліки автоматизованого тестування UX/UI дизайну з використанням методів штучного інтелекту.

Мета дослідження.

Оцінити обмеження автоматизованого тестування UX/UI дизайну на основі методів штучного інтелекту.

Результати дослідження.

Недоліки автоматизованого тестування UX/UI на основі ШІ можна поділити на категорій:

1. Обмеженість моделей ШІ. Більшість сучасних моделей, хоча й мають високий рівень точності в обробці великих обсягів даних, не завжди здатні точно імітувати людські емоції та інтуїтивні реакції на інтерфейс. Це може призвести до хибних висновків про зручність використання інтерфейсу[2].

2. Невизначеність у перевірці нових або нестандартних інтерфейсів. ШІ моделі часто надають перевагу перевірці стандартних елементів дизайну, не здатні ефективно працювати з інноваційними або нестандартними інтерфейсами, що може призвести до недооцінки важливих аспектів дизайну[3].

3. Виникнення хибнопозитивних результатів. Автоматизовані системи можуть не враховувати всі контекстні взаємодії користувача з інтерфейсом, що призводить до неправильних результатів тестування[4].

4. Обмеженість у сприйнятті мультимодальних елементів. UX/UI дизайн включає не лише візуальні, а й аудіо. ШІ може не в змозі повноцінно оцінити мультимодальні аспекти інтерфейсу, що важливо для деяких категорій користувачів, наприклад, для людей з обмеженими можливостями[5].

5. Висока вартість впровадження. Для інтеграції та налаштування систем на базі ШІ потрібні значні ресурси, в тому числі для підготовки даних та налаштування алгоритмів. Це може бути економічно невиправданим для стартапів та проектів які не мають великих фінансових можливостей[4].

Висновки та перспективи.

Автоматизоване тестування UX/UI дизайну з використанням методів штучного інтелекту стикається з недоліками, які потребують подальшого дослідження та розвитку нових методів тестування.

Моделі ШІ не завжди здатні коректно враховувати емоційні та інтуїтивні аспекти взаємодії користувача, що обмежує їхню точність у оцінці загального користувацького досвіду.

Врахування недоліків дозволить значно покращити якість тестування та забезпечити кращий користувацький досвід.

Недоліки моделей ШІ залишатимуться актуальними, що вказує на необхідність постійного залучення людини до процесу перевірки результатів у тій чи іншій мірі.

Список використаних джерел:

1. Alexander, P., & Thompson, K. (2021). *Artificial Intelligence in UX/UI Testing: Challenges and Opportunities*. *Journal of Software Testing*, 15(3), 45-59.
2. Lee, J., & Hwang, S. (2020). *Automation of User Experience Testing Using Deep Learning*. *International Journal of Human-Computer Interaction*, 36(7), 615-628.
3. Miller, R., & Scott, D. (2022). *Machine Learning Models in UX Design Evaluation*. *Journal of Interaction Design*, 12(4), 134-145.
4. , A., & Garcia, M. (2019). *Evaluating the Effectiveness of AI-Based UX/UI Testing Tools*. *IEEE Transactions on Software Engineering*, 45(5), 78-89.
5. Watson, T., & Kim, H. (2023). *Deep Learning for User Interface Design: A New Frontier in Automated Testing*. *Computers in Human Behavior*, 78(2), 110-121.

Техніки тест-дизайну для підвищення ефективності тестування програмного забезпечення

*Безугла Вікторія Юріївна,
студентка групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
vuv37783@gmail.com*

*Науковий керівник: Щербина Ірина Сергіївна,
кандидат технічних наук,
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Тестування програмного забезпечення (ПЗ) є критично важливим етапом у процесі розробки, що дозволяє забезпечити відповідність ПЗ вимогам користувачів та високий рівень його якості. Використання сучасних технік тест-дизайну дає змогу

значно підвищити ефективність тестування, зменшуючи витрати часу та кількість помилок.

Постановка задачі.

Основним завданням є опис і визначення практичної цінності наступних технік тест-дизайну: еквівалентного розбиття, аналізу граничних значень, тестування на основі діаграм станів, приймального тестування та парного тестування.

Мета дослідження.

Аналіз ключових технік тест-дизайну, які сприяють оптимізації тестового процесу та підвищенню його точності.

Результати дослідження.

Основні техніки тест-дизайну

1. Еквівалентне розбиття

Метод еквівалентного розбиття передбачає поділ вхідних даних на класи еквівалентності. Це дозволяє мінімізувати кількість тестових випадків, не знижуючи їх ефективності. Перевірка кожного класу дає змогу зосередитися на ключових характеристиках даних, зменшуючи ймовірність пропуску дефектів [1, с. 53].

2. Аналіз граничних значень

Ця техніка спрямована на перевірку роботи системи в межах крайніх значень вхідних даних. Помилки найчастіше виникають саме на межах діапазонів, тому тестування таких значень є одним із найефективніших способів виявлення дефектів [2, с. 64].

3. Тестування за допомогою діаграм станів

Використання діаграм станів забезпечує моделювання поведінки системи залежно від змін її станів. Цей метод ефективний для складних програмних систем із багатьма залежностями, дозволяючи виявляти помилки в логіці переходів [3, с. 29].

4. Приймальне тестування

Метод базується на перевірці функціональності системи відповідно до реальних вимог користувачів. Це дозволяє оцінити готовність продукту до реального використання, що безпосередньо впливає на задоволеність кінцевих користувачів [4, с. 88].

5. Парне тестування

Парне тестування (Pairwise Testing) передбачає перевірку всіх можливих пар значень вхідних змінних. Це дозволяє зменшити загальну кількість тестів, одночасно забезпечуючи високу ймовірність виявлення дефектів, що виникають у взаємодії різних параметрів [5, с. 42].

Висновки та перспективи.

Використання технік тест-дизайну дозволяє підвищити ефективність тестування програмного забезпечення, мінімізувати кількість дефектів та оптимізувати затрати часу. Кожен із методів має свої переваги, що дозволяє тестувальникам обирати найефективніші підходи залежно від потреб проєкту.

Список використаних джерел:

1. Бабічев С. П. Основи тестування ПЗ. Київ: Технополіс, 2021. 276 с.
2. Литвиненко І. В. Техніки аналізу в тестуванні. Харків: Ранок, 2020. 324 с.
3. Джонсон К. Тестування складних систем. Львів: Фенікс, 2019. 210 с.
4. Smith J. Advanced Software Testing. 3rd ed. London: TechWorld, 2018. 470 p.
5. Lee M., Chang D. Efficient Pairwise Testing. New York: CodePress, 2019. 190 p.

Захист фронтенд-компонентів від XSS-атак за допомогою Content Security Policy

*Білодід Даніель Вадимович,
студент 6 курсу, групи ПДМ-63
Державного університету інформаційно-комунікаційних технологій
danielbeloded@gmail.com*

*Шахматов Іван Олександрович
викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій
ivan.shakhmatov@gmail.com*

*Науковий керівник: Довженко Тимур Павлович,
к.т.н., доцент кафедри Інженерії програмного забезпечення,
Державний університет інформаційно-комунікаційних технологій*

Постановка задачі.

Однією з найсерйозніших загроз для веб-додатків є атаки типу Cross-Site Scripting (XSS), де зловмисники впроваджують шкідливий код у сторінки додатка, отримуючи доступ до конфіденційної інформації користувачів. Це може призвести до витоку даних, зміни поведінки додатка, або навіть до компрометації облікових записів користувачів. Проблема забезпечення безпеки фронтенд-компонентів є особливо актуальною, оскільки більшість сучасних веб-додатків значною мірою використовують JavaScript і інші динамічні технології для інтерфейсу користувача, що збільшує можливості для атак XSS. Content Security Policy (CSP) пропонує ефективний механізм захисту від цих атак, дозволяючи встановлювати політики безпеки для обмеження виконання підозрілого коду в браузері.

Мета дослідження.

Метою дослідження є аналіз можливостей CSP як ефективного інструменту захисту фронтенд-компонентів від шкідливого коду, а також дослідження специфічних стратегій і налаштувань CSP, які допоможуть мінімізувати ризики атак XSS та інших вразливостей. Дослідження також спрямоване на порівняння ефективності CSP з традиційними методами захисту та виявлення його переваг і обмежень у різних контекстах веб-додатків.

Практична реалізація.

Першим кроком є встановлення npm-паketу Express, який служить легким фреймворком для створення серверних додатків на Node.js. Після цього необхідно встановити пакет helmet, який допомагає захистити додаток шляхом налаштування відповідних HTTP-заголовків, зокрема CSP. Налаштування CSP здійснюється за допомогою об'єкта directives, в якому визначаються правила для різних типів ресурсів.

Директива default-src встановлює базові джерела для завантаження ресурсів за замовчуванням; у цьому випадку дозволяються ресурси лише з того ж домену, що і сам додаток.

Директива script-src визначає дозволені джерела для завантаження JavaScript-кодів, дозволяючи скрипти з того ж домену та з довірених CDN. Директива style-src обмежує джерела стилів, контролюючи візуальне оформлення додатку. Директива img-src дозволяє завантажувати зображення з того ж домену та з data-URI, що знижує ризик впровадження шкідливих зображень, оскільки data-URI забезпечує контрольоване джерело даних у самому коді додатку. Директива object-src забороняє використання об'єктів, таких як Flash, які можуть бути вразливими до атак. Нарешті, директива upgradeInsecureRequests автоматично оновлює всі HTTP-запити до HTTPS, забезпечуючи, щоб усі ресурси завантажувалися через захищене HTTPS-з'єднання. Для інтеграції CSP у HTML-код додатку використовується мета-тег, який містить налаштування політики безпеки. Це дозволяє браузеру застосовувати встановлені правила при завантаженні сторінки.

Після впровадження CSP необхідно протестувати додаток у браузері. Відкривши консоль розробника, можна перевірити наявність повідомлень про помилки, які вказують на блокування ресурсів CSP. Це допоможе ідентифікувати та виправити потенційні проблеми з забороненими ресурсами. Впровадження CSP у веб-додатки на базі Express з використанням пакету helmet є ефективним способом підвищення безпеки та запобігання впровадженню шкідливого коду у фронтенд-компоненти. Налаштування відповідних директив дозволяє контролювати джерела завантаження

ресурсів, що значно знижує ризик атак типу XSS та інших вразливостей, пов'язаних з безпекою вмісту. Зазначені операції можуть бути представлені у вигляді програмних модулів (Рис.1).

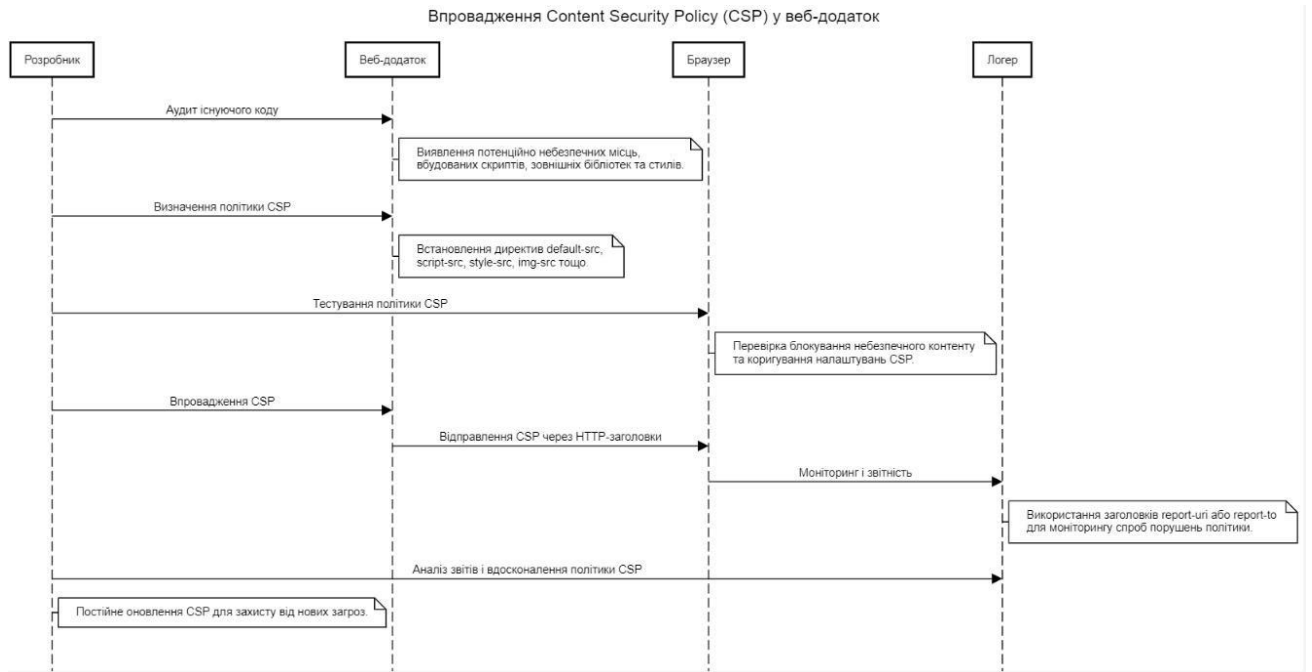


Рис.1 Діаграма впровадження Content Security Policy (CSP) у веб-додаток

Результати дослідження.

Результати показують, що впровадження CSP значно знижує ризик XSS-атак завдяки контролю над джерелами контенту, який може завантажуватися та виконуватися в браузері. CSP дозволяє веб-додатку вказувати, які саме ресурси (скрипти, стилі, зображення, медіафайли) дозволено завантажувати з довірених джерел, блокуючи всі інші. Наприклад, використання налаштувань таких директив CSP, як `default-src`, `script-src`, `img-src`, дозволяє забезпечити надійний захист, вказавши, з яких серверів дозволено завантажувати певні ресурси. Такий підхід значно знижує ймовірність впровадження небажаного JavaScript-коду, який може викликати небажані дії на сторінці.

Практичне застосування CSP виявилось ефективним на прикладах популярних веб-додатків. Активна CSP-стратегія дозволила блокувати шкідливий контент до його виконання в браузері. Використання політики Content-Security-Policy: script-src 'self'; дозволяє виконувати скрипти лише з поточного домену, що забезпечує захист від шкідливого JavaScript-коду, який міг бути впроваджений з зовнішніх джерел. Крім того, налаштування CSP сприяє обмеженню інших потенційно небезпечних елементів, таких як вставки вбудованих iFrame або стилів, які можуть також використовуватися для атак.

Висновки та перспективи.

Використання CSP є важливим та ефективним методом забезпечення безпеки фронтенд-компонентів веб-додатків. Дослідження показало, що належне впровадження CSP дозволяє ефективно обмежувати джерела виконання коду і значно знижує ризик атак типу XSS. З розвитком інтернет-загроз CSP стає обов'язковим інструментом безпеки для сучасних веб-додатків, надаючи веб-розробникам додатковий рівень контролю. У подальшому дослідження перспектив інтеграції CSP з іншими механізмами захисту (наприклад, багаторівневими системами аутентифікації) може забезпечити ще більшу надійність захисту додатків.

Перспективи застосування CSP також включають розробку нових директив та можливостей для більш детального контролю за ресурсами і способами їх завантаження, що розширить сферу його застосування й забезпечить надійний захист від зростаючої кількості інтернет-загроз.

Список використаних джерел:

1. Okusi, O. Cyber Security Techniques for Detecting and Preventing Cross-Site Scripting Attacks // World Journal of Innovation and Modern Technology. – 2024. – Vol. 8, No. 2. – С. 71–89. – DOI: 10.56201/wjimt.v8.no2.2024.pg71.89.
2. Stamm, Sid, Brandon Sterne, Gervase Markham. Reining in the web with content security policy // Proceedings of the 19th international conference on World wide web. – ACM, 2010.

- [Електронний ресурс]. – Режим доступу: <https://dl.acm.org/doi/10.1145/1772690.1772702>.
3. Mozilla Developer Network (MDN). Content Security Policy (CSP). – [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>.
4. Weamie, S. Cross-Site Scripting Attacks and Defensive Techniques: A Comprehensive Survey // International Journal of Communications, Network and System Sciences. – 2022. – Vol. 15. – С. 126–148. – DOI: 10.4236/ijcns.2022.158010.
5. Abu Al-Haija, Q. Cost-effective detection system of cross-site scripting attacks using hybrid learning approach // Results in Engineering. – 2023. – Vol. 19. – Article ID: 101266. – [Електронний ресурс]. – Режим доступу: <https://doi.org/10.1016/j.rineng.2023.101266>.
6. Alsaffar, M., Aljaloud, S., Mohammed, B. A., Al-Mekhlafi, Z. G., Almurayziq, T. S., Alshammari, G., Alshammari, A. Detection of Web Cross-Site Scripting (XSS) Attacks // Electronics. – 2022. – Vol. 11, No. 14. – Article ID: 2212. – DOI: 10.3390/electronics11142212.

Інтеграція метрик відстежуваності та поведінкових метрик для оптимізації рефакторингових рішень у програмних системах

*Кулаков Олексій Віталійович,
аспірант групи АІПЗ-11
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
alexeyvk9@gmail.com*

*Науковий керівник: Замрій Ірина Вікторівна,
д.т.н., доцент, завідувач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Рефакторинг програмного забезпечення є основним елементом забезпечення його довготривалої підтримуваності та зручності для розробників. Існуючі методи рефакторингу здебільшого зосереджені на оцінці архітектурної якості коду через метрики когезії та залежності, що не враховують впливу на продуктивність і

підтримуваність системи [2][3]. Два нових аспекти рефакторингу — метрики відстежуваності вимог і поведінкові метрики, можуть забезпечити комплексне розуміння системи після внесення змін.

Постановка задачі

У ході дослідження поставлено завдання:

1. Проаналізувати існуючі підходи до рефакторингу програмного забезпечення, зокрема використання архітектурних метрик когезії та залежності, їхні обмеження та вплив на якість коду.
2. Вивчити можливості інтеграції метрик відстежуваності вимог для покращення зрозумілості та впорядкованості зв'язків між вимогами і кодом у процесі рефакторингу.
3. Оцінити перспективи застосування поведінкових метрик (час виконання функцій, використання пам'яті та процесора) для збереження продуктивності системи після рефакторингу.

Задачі дослідження спрямовані на визначення шляхів покращення процесу рефакторингу шляхом аналізу впливу архітектурних, відстежувальних та поведінкових метрик, що базується на даних з проаналізованих джерел.

Мета дослідження.

Метою дослідження є аналіз можливостей та перспектив застосування архітектурних, відстежувальних та поведінкових метрик для покращення процесу рефакторингу програмного забезпечення. Дослідження спрямоване на вивчення їх впливу на архітектурну якість, продуктивність та підтримуваність системи.

Результати дослідження.

У рамках поставлених задач було проведено аналіз можливостей і перспектив застосування архітектурних, відстежувальних та поведінкових метрик для покращення процесу рефакторингу програмного забезпечення[1][2]. Рефакторинг зосереджувався на оптимізації зв'язків між компонентами системи та підвищенні ефективності роботи коду. Основні результати наведені нижче.

Визначення ключових термінів

1. Когезія — це міра зв'язаності методів і атрибутів у межах одного класу. Висока когезія вказує на те, що клас виконує єдину функцію, що підвищує його зрозумілість і легкість у підтримці[3, 4]. Формула когезії:

$$Coh = \frac{m \cap a}{\max(m, a)}$$

де m - кількість методів у класі, a — кількість атрибутів.

2. Залежність - це міра взаємозалежності між класами системи. Висока залежність ускладнює внесення змін, оскільки зміни в одному класі можуть спричинити зміни в інших[5].

Формула залежності:

$$Dependency = \frac{|E_{external}|}{|E_{total}|}$$

де $E_{external}$ - кількість зовнішніх зв'язків класу, E_{total} — загальна кількість зв'язків.

3. Ентропія відстежуваності — це показник, що характеризує рівень впорядкованості зв'язків між вимогами та кодом[3]. Низька ентропія вказує на високу впорядкованість, що полегшує підтримку коду.

$$H(T) = - \sum_{i=1}^n p_i \log(p_i)$$

де p_i — ймовірність зв'язку між вимогою та елементом коду, а n – загальна кількість зв'язків. Висока ентропія свідчить про високий рівень неупорядкованості, що ускладнює підтримку системи.

4. По ведінкові метрики[3] — це показники, що оцінюють продуктивність програмного забезпечення. Включають:

$$Texes = \frac{1}{n} \sum_{i=1}^n t_i$$

Де T_{exec} - час виконання функцій, t_i — час виконання окремого методу, а n — загальна кількість методів. Скорочення часу виконання свідчить про покращення ефективності, тоді як його збільшення може свідчити про погіршення продуктивності.

Аналіз архітектурних метрик

Суть рефакторингу: Рефакторинг включав редагування коду для зменшення надмірної взаємозалежності між класами та покращення когезії всередині класів.

Це було досягнуто шляхом:

- Розбиття великих класів із низькою когезією на кілька більш спеціалізованих.
- Усунення циклічних залежностей між класами.
- Оптимізації ієрархії успадкування для зменшення складності зв'язків між компонентами.

Код, який редагувався:

- До рефакторингу: У системі були класи, які виконували кілька несуміжних завдань, що знижувало їх зрозумілість. Наприклад, клас `DataProcessor` поєднував логіку обробки даних із функціями збереження даних у базу.
- Після рефакторингу: Функціональність класу `DataProcessor` була розділена на два окремі класи: `DataTransformer` для обробки даних та `DatabaseSaver` для їх збереження.

Результати оцінки метрик:

- До рефакторингу: $Coh=0.45$, $Dep=0.70$.
- Після рефакторингу: $Coh=0.70$, $Dep=0.44$.

Це свідчить про підвищення зв'язаності методів і атрибутів у класах та зменшення кількості зовнішніх зв'язків.

Інтеграція метрик відстежуваності

Суть рефакторингу: Враховувалася необхідність підтримки зв'язків між вимогами та кодом, щоб забезпечити зрозумілість змін для розробників. Це включало:

- Перепроектування модулів для збереження логічних зв'язків між вимогами і їх реалізацією.
- Усунення надлишкових залежностей, які ускладнювали відстеження.

Код, який редагувався:

- До рефакторингу: Код мав складні зв'язки між модулями, наприклад, функція `generateReport()` викликала методи з кількох непов'язаних модулів, що створювало хаос у відстежуваності вимог.
- Після рефакторингу: Функція була розбита на кілька вузькоспеціалізованих функцій, кожна з яких виконувала окреме завдання (наприклад, отримання даних, обробка даних, створення звіту).

Результати:

- До рефакторингу: $H(T)=0.60H(T)$.
- Після рефакторингу: $H(T)=0.38H(T)$.

Це вказує на покращення впорядкованості зв'язків між вимогами і кодом.

Оцінка поведінкових метрик

Суть рефакторингу: Вплив змін на продуктивність оцінювався шляхом аналізу найбільш ресурсомістких операцій. Для зменшення часу виконання та використання пам'яті було:

- Оптимізовано алгоритми у функціях.
- Зменшено кількість дублювання обчислень у циклах.

Код, який редагувався:

- До рефакторингу: Алгоритми мали значне дублювання коду, наприклад, функція `calculateStatistics()` виконувала повторні обчислення для кожного запиту, що впливало на продуктивність.
- Після рефакторингу: Введено кешування проміжних результатів для уникнення повторного виконання тих самих обчислень.

Як результат час виконання функції (ТехесТ) зменшився:

- До рефакторингу: Техес=510 мс.
- Після рефакторингу: Техес=450 мс (зниження на 12%).

Висновки та перспективи

Результати дослідження підтвердили, що інтеграція архітектурних, відстежувальних та поведінкових метрик у процес рефакторингу є ефективним підходом до покращення якості програмного забезпечення. Аналіз показав, що використання метрик когезії та залежності дозволяє оптимізувати архітектурну структуру коду, знижуючи надмірні залежності між класами та підвищуючи внутрішню зв'язаність їх методів і атрибутів. Включення метрик відстежуваності забезпечує збереження логічних зв'язків між вимогами та кодом, знижуючи ентропію зв'язків і підвищуючи зрозумілість системи для розробників. Це особливо важливо для довготривалої підтримованості програмного забезпечення.

Оцінка поведінкових метрик показала, що рефакторинг може не лише покращити архітектурні характеристики, але й зберегти продуктивність системи. Оптимізація алгоритмів, усунення дублювання коду та впровадження кешування сприяли зниженню часу виконання функцій, використання пам'яті та завантаженості процесора. Це підкреслює важливість інтеграції поведінкових метрик для комплексного оцінювання впливу рефакторингу.

Таким чином, запропонований підхід дозволяє не лише усунути технічний борг і підвищити якість коду, але й зробити систему більш підтримуваною та адаптивною до змін. Отримані результати демонструють перспективність подальшого

дослідження та впровадження такого підходу в сучасних процесах розробки складних програмних систем.

Список використаних джерел:

1. Priyadarshni Suresh Sagar, Eman Abdullah Alomar, Eman Abdullah Alomar, Mohamed Wiem Mkaouer, Mohamed Wiem Mkaouer, “Comparing Commit Messages and Source Code Metrics for the Prediction of Refactoring Activities”, September 2021, Algorithms 14(10):289, DOI: 10.3390/a14100289.
2. Ally S. Nyamawe, Hui Liu, Zhendong Niu, Wentao Wang, Nan Niu, “Recommending Refactoring Solutions Based on Traceability and Code Metrics”, IEEE Access, Page(s): 49460 - 49475, 06 September 2018, Electronic ISSN: 2169-3536, DOI: 10.1109/ACCESS.2018.2868990.
3. Umar Iftikhar, Umar Iftikhar, Nauman Bin Ali, Nauman Bin Ali, Jürgen Börstler, Jürgen Börstler, Muhammad Usman, “A Catalog of Source Code Metrics – A Tertiary Study”, Software Quality: Higher Software Quality through Zero Waste Development, May 2023
DOI: 10.1007/978-3-031-31488-9_5.
4. Umar Iftikhar, Nauman Bin Ali, Jürgen Börstler, Muhammad Usman, “A tertiary study on links between source code metrics and external quality attributes”, Information and Software Technology, Volume 165, January 2024, 107348
5. D. St^oahl, A. Martini, and T. M^oartensson, “Big bangs and small pops: On critical cyclomatic complexity and developer integration behavior,” in 2019 IEEE/ACM 41st International Conference on Software Engineering: (ICSE-SEIP), 2019.

Оптимізація процесу розробки програмного забезпечення шляхом впровадження алгоритму інкрементної обробки файлів

*Фетько Юрій Іванович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
fetkoюра@gmail.com
Науковий керівник: Залива Віталій Вікторович,
доктор філософії (PhD),
ст.викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

В сучасному процесі розробки програмного забезпечення критичним фактором є швидкість та ефективність обробки файлів під час збірки проекту. Особливо це стосується веб-розробки, де постійно зростаючий обсяг CSS та JavaScript файлів створює значне навантаження на процес розробки. Традиційні підходи до обробки файлів часто призводять до надмірного використання ресурсів та збільшення часу розробки [2-4].

Постановка задачі.

Розробка та впровадження алгоритму інкрементної обробки файлів для оптимізації процесу веб-розробки з використанням системи збірки Gulp [1].

Мета дослідження.

Підвищення ефективності процесу розробки програмного забезпечення шляхом оптимізації обробки CSS та JavaScript файлів за допомогою селективного підходу та інтелектуального кешування.

Результати дослідження.

Розроблений алгоритм базується на принципі селективної обробки файлів з використанням графа залежностей. Основними компонентами системи є:

1. Механізм аналізу залежностей між файлами
2. Система інтелектуального кешування
3. Селективна обробка модифікованих файлів

4. Паралельна обробка незалежних компонентів

Алгоритм впроваджує механізм відстеження змін файлів на основі хешування та аналізу залежностей, що дозволяє обробляти лише модифіковані файли та їх залежності. Це суттєво знижує навантаження на систему та прискорює процес розробки [5].

За результатами тестування, впровадження розробленого алгоритму дозволило:

- Скоротити час збірки проекту на 30-40%
- Зменшити використання системних ресурсів на 20-30%
- Оптимізувати процес розробки за рахунок миттєвого оновлення змінених файлів

Висновки та перспективи.

Розроблений алгоритм демонструє значне підвищення ефективності процесу розробки програмного забезпечення. Використання селективного підходу та інтелектуального кешування дозволяє суттєво скоротити час обробки файлів та оптимізувати використання системних ресурсів.

Подальші дослідження можуть бути спрямовані на розширення функціоналу алгоритму для підтримки інших типів файлів та інтеграції з іншими системами збірки.

Список використаних джерел:

1. Gulp Documentation. [Електронний ресурс]. Режим доступу: <https://gulpjs.com/docs/en/getting-started/quick-start>
2. Григорик, І. "High Performance Browser Networking: What every web developer should know about networking and web performance", O'Reilly Media, 2023. - 286 с.
3. Османі, А. "Learning Patterns: JavaScript Design Patterns, Testing and Performance Optimization", O'Reilly Media, 2023. - 412 с.
4. Node Package Manager (NPM) - документація українською [Електронний ресурс] // Node.js. – 2023. – Режим доступу: <https://nodejs.org/uk/docs>
5. Шивакумар Ш. К. "Modern Web Performance Optimization: Methods, Tools, and Patterns to Speed Up Digital Platforms" / Шайлеш Кумар Шивакумар; пер. з англ. О. Петренка. – К.: Видавничий дім "КМ Академія", 2023. – 380 с.

Адаптивні алгоритми перевірки знань у вивченні мови програмування JAVA

*Шевчук Юрій Олександрович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
yuriishevchuk10@gmail.com
Науковий керівник: Замрій Ірина Вікторівна,
доктор технічних наук, доцент,
завідувач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Впровадження адаптивних алгоритмів у процес перевірки знань з мови програмування Java стає все більш актуальним питанням сучасної технічної освіти. Такі алгоритми дозволяють створювати індивідуальні траєкторії навчання та оцінювання для кожного студента, підвищуючи ефективність освітнього процесу [1, с. 45].

Постановка задачі.

Дослідити та розробити адаптивні алгоритми для автоматизованої перевірки знань з мови програмування Java, які враховують індивідуальні особливості та рівень підготовки студентів.

Мета дослідження.

Підвищення ефективності процесу оцінювання знань студентів з мови Java шляхом впровадження адаптивних алгоритмів тестування.

Результати дослідження.

Адаптивне тестування базується на принципі динамічного підбору завдань відповідно до рівня знань студента. Система починає з завдань середнього рівня складності та, залежно від відповідей, коригує складність наступних питань. Це дозволяє більш точно визначити рівень знань та зменшити час тестування [2, с. 67].

Розроблений алгоритм складається з чотирьох основних компонентів. Перший компонент - це банк завдань різного рівня складності, який містить широкий спектр

питань та практичних завдань з Java, розподілених за рівнями складності та тематичними розділами. Другий компонент представляє собою систему оцінки складності кожного завдання, яка використовує комбінацію експертних оцінок та статистичного аналізу результатів попередніх тестувань для точного визначення рівня складності. Третій компонент - це механізм адаптивного підбору наступного завдання, який базується на аналізі відповідей студента та динамічно коригує рівень складності наступних питань. Четвертий компонент - алгоритм розрахунку фінальної оцінки, який враховує не лише кількість правильних відповідей, але й складність виконаних завдань та час, витрачений на їх вирішення.

Цінною особливістю системи є використання методів машинного навчання для аналізу патернів відповідей студентів та уточнення оцінки складності завдань [3, с. 112].

Система також включає модуль аналітики, який надає викладачам детальну інформацію про типові помилки студентів та проблемні теми, що дозволяє оптимізувати навчальний процес [4, с. 89].

Помітним аспектом є автоматична генерація індивідуальних рекомендацій для кожного студента щодо тем, які потребують додаткового опрацювання. Це допомагає створити персоналізований план навчання та підвищити ефективність самостійної роботи.

Висновки і пропозиції.

Впровадження адаптивних алгоритмів тестування значно підвищує ефективність оцінювання знань з мови Java. Система забезпечує більш точну оцінку рівня знань студентів при меншвих витратах часу. Подальший розвиток системи може включати розширення банку завдань, вдосконалення алгоритмів адаптації та інтеграцію з існуючими системами управління навчанням.

Список використаних джерел

1. Іванов А. В. Адаптивне тестування в освіті: теорія та практика : монографія. Київ : Освіта України, 2022. 180 с.
2. Коваленко Т. М. Машинне навчання в освітніх системах : навч. посіб. Харків : НТУ "ХП", 2023. 245 с.
3. Петров С. О. Сучасні методи оцінювання знань у програмуванні : підручник. Львів : Видавництво Львівської політехніки, 2022. 198 с.
4. Сидоренко О. В. Автоматизовані системи контролю знань : монографія. Київ : КПІ ім. Ігоря Сікорського, 2023. 156 с.

Розробка методу оптимізації програмного коду на основі генетичного алгоритму

*Шоробура Владислав Вікторович
студент групи ПДМ-62
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
sharik201620152014@gmail.com
Науковий керівник: Худік Богдан Олександрович,
доктор філософії (PhD),
ст. викладач кафедри Інженерії програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій*

Генетичні алгоритми (ГА) широко використовуються для вирішення задач оптимізації завдяки своїй здатності моделювати природний еволюційний процес. Це дозволяє створювати й комбінувати різноманітні рішення, поступово їх вдосконалюючи на основі заданого критерію придатності. У сфері оптимізації програмного коду такі алгоритми сприяють генерації та вибору найпродуктивніших варіантів.

Проте, попри наявні дослідження, ефективне застосування ГА для автоматизованої оптимізації коду залишається відкритим питанням. Існуючі методи здебільшого орієнтовані на вузькі аспекти, як-от підвищення продуктивності окремих фрагментів коду, але не враховують комплексної оптимізації всієї системи. Це особливо актуально для України, де зростає попит на ефективне програмне забезпечення в різних секторах.

Оптимізація коду є ключовою задачею сучасного програмування, адже продуктивність програмного забезпечення прямо впливає на його швидкодію, ефективність використання ресурсів і масштабованість. Зі збільшенням обсягів даних, ускладненням програмних систем і необхідністю раціонального використання обчислювальних ресурсів значущість оптимізаційних методів лише зростає. Серед цих методів ГА займають важливе місце, оскільки дозволяють знаходити глобальні оптимальні рішення та працювати з великою кількістю варіантів, що відкриває нові можливості для вдосконалення коду без втрати його функціональності.

Генетичні алгоритми (ГА) є потужним методом оптимізації, що базується на принципах природного добору для пошуку оптимальних рішень. Ці алгоритми імітують еволюційний процес, поступово вдосконалюючи покоління можливих розв'язків шляхом використання біологічно натхненних операцій, таких як селекція, схрещування та мутація. ГА довели свою ефективність у розв'язанні складних задач оптимізації, особливо тих, які мають великий простір можливих рішень або в яких традиційні методи не дають задовільних результатів.

Основні принципи роботи генетичних алгоритмів подані в табл. 1. [1].

Основні принципи роботи генетичних алгоритмів

№	Принципи	Пояснення
1	2	3
1	Імітація біологічної еволюції	ГА базується на концепції еволюції Чарльза Дарвіна, яка передбачає природний добір найбільш пристосованих особин для виживання. У ГА цей принцип застосовується для пошуку найкращих рішень шляхом «еволюції» набору можливих рішень або «популяції». Популяція складається з численних «особин», кожна з яких представляє можливий розв'язок задачі. Метою ГА є поступове вдосконалення популяції шляхом повторного застосування алгоритмічних кроків
2	Кодування розв'язків у вигляді генетичних представлень	У генетичних алгоритмах розв'язки задачі кодуються зазвичай у вигляді рядків з чисел, бітів або символів, які називають «хромосомами». Кожен елемент хромосоми, що відповідає певній характеристиці розв'язку, називається «геном». Вибір схеми кодування є ключовим для успішного функціонування ГА, оскільки вона визначає можливість і точність оцінки кожного рішення
3	Оцінка пристосованості рішень	У ГА кожен розв'язок оцінюється за допомогою спеціальної функції — "функції пристосованості", яка визначає, наскільки добре цей розв'язок відповідає критеріям задачі. Функція пристосованості відіграє вирішальну роль у відборі рішень для подальшої оптимізації. Ті рішення, які мають вищі значення функції пристосованості, отримують більшу ймовірність бути обраними для наступного покоління, що забезпечує поступове покращення популяції

Генетичні алгоритми — це універсальний інструмент, який ефективно вирішує складні задачі оптимізації, моделюючи процеси біологічної еволюції. Основи ГА охоплюють ключові концепції, такі як кодування рішень, оцінка їхньої пристосованості, а також базові операції: відбір, схрещування та мутація. Завдяки цим механізмам алгоритми здатні адаптуватися та генерувати нові варіанти рішень, що забезпечує високу ефективність у випадках, коли традиційні методи виявляються малопродуктивними. Такий підхід дозволяє досягати кращої продуктивності та точності рішень, що робить ГА перспективним інструментом для вдосконалення програмного забезпечення та оптимізації коду.

Список використаних джерел:

1. Бажан В. Використання генетичних алгоритмів із застосуванням випадкових процесів при розв'язуванні задач оптимізації. November 2023. ГРААЛЬ НАУКИ
2. Kannadasan R., Manoj Kumar K. N., Sistla K. Code Optimization using Genetic Algorithm/ November 2016SSRN Electronic Journal
3. Cheng, L., & Lee, H. (2010). "Enhanced Genetic Algorithms for Code Optimization." ACM Transactions on Software Engineering and Methodology.

Методика UX-оптимізації веб-ресурсів для підвищення залученості користувачів в ігрових онлайн-проектах

Юхта Максим Артемович

студент групи ПДМ-63

спеціальність 121 Інженерія програмного забезпечення

Державного університету інформаційно-комунікаційних технологій

BezNika2012@gmail.com

Науковий керівник: Компан Сергій Володимирович

к.ф.-м.н, доцент кафедри Інтернет-технологій

Державного університету інформаційно-комунікаційних технологій

Ігрові веб-платформи є важливими точками контакту між гравцями та розробниками. Вони забезпечують доступ до інформації про гру, спільноти та додаткові послуги, такі як завантаження чи покупки. В умовах конкуренції в індустрії ігор стає критично важливим не лише створити естетичний дизайн, а й забезпечити зручність, швидкість та ефективність взаємодії з користувачами. Для цього застосовуються сучасні аналітичні системи, які допомагають вивчати поведінку гравців на платформі та впроваджувати зміни, що покращують їхній досвід.

Постановка задачі.

Задачею є розробка методики UX-оптимізації веб-ресурсів, яка враховує потреби користувачів ігрових проєктів, їхню поведінку на платформі та ключові фактори залученості. Така методика має покращити досвід взаємодії, підвищити час перебування користувачів на платформі та знизити відсоток відмов.

Мета дослідження.

Мета полягає у визначенні основних принципів UX-оптимізації, які дозволяють створити зручний та інтерактивний інтерфейс веб-ресурсів для ігрових проєктів, а також у впровадженні інструментів для аналізу поведінки користувачів та оцінки ефективності запропонованих змін.

Результати дослідження.

Запропонована методика базується на застосуванні сучасних технологій і практик UX-дизайну, які включають: Гейміфікацію, тобто інтеграцію таких елементів, як нагороди, досягнення або рейтинги, які стимулюють активну участь користувачів. Наприклад, додавання системи "ачівок" за виконання певних дій на платформі (реєстрація, коментарі, перегляд відео) підвищило кількість взаємодій на 27%; Адаптивний дизайн, тобто забезпечення коректного відображення платформи на різних пристроях, що дозволило збільшити частку мобільних користувачів на 15%; Аналіз поведінки, тобто використання інструментів, таких як Hotjar і Google Analytics, для вивчення маршрутів користувачів і виявлення проблемних зон на сайті.

Після впровадження UX-оптимізації час завантаження сайту було знижено в середньому на 1,5 секунди, що зменшило відсоток відмов на 18%. Крім того, реорганізація структури меню дозволила спростити доступ до популярних функцій, таких як завантаження гри чи контактна підтримка, що позитивно вплинуло на задоволеність користувачів.

Розробка системи персоналізації на основі даних користувачів дозволила створити динамічний головний екран, який відображає найбільш релевантний контент залежно від інтересів гравця. Це сприяло підвищенню рівня кліків (CTR) на 22%.

Висновки та перспективи.

Методика UX-оптимізації довела свою ефективність у підвищенні залученості користувачів на веб-ресурсах ігрових проєктів. Запроваджені підходи, такі як гейміфікація, адаптивний дизайн та персоналізація, сприяли поліпшенню ключових показників ефективності платформи.

Подальші дослідження можуть бути спрямовані на інтеграцію алгоритмів штучного інтелекту для автоматичного аналізу даних і адаптації UX-дизайну залежно від змін у поведінці користувачів у реальному часі.

Список використаних джерел:

1. Usability Geek - Usability Geek. – [Електронний ресурс] – Режим доступу: <https://usabilitygeek.com/achieving-better-engagement-through-ux-the-secret-to-gamification/>.
2. Using gamification to increase website engagement. Abmatic AI | Transforming Account-Based Marketing. – [Електронний ресурс] – Режим доступу: <https://abmatic.ai/blog/gamification-to-increase-website-engagement>.
3. UX Optimization: Everything You Need to Know in 2024. Survicate: Effortless Survey Software. – [Електронний ресурс] – Режим доступу: <https://survicate.com/blog/ux-optimization/>.
4. Ways to Use Gamification in UX Design | inVerita. inVerita. – [Електронний ресурс] – Режим доступу: <https://inveritasoft.com/article-ways-to-use-gamification-in-ux-design>.

Врахування ризиків та адаптивності при визначенні ефективності моделі прогнозування поведінки фінансових ринків

*Скидан Павло Валерійович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
pavelskidan2020@gmail.com
Науковий керівник: Замрій Ірина Вікторівна,
д.т.н, доцент, завідувач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Сучасні фінансові ринки характеризуються високим рівнем змін та складною динамікою, що вимагає використання адаптивних моделей для точного прогнозування. Більшість існуючих моделей оцінки ефективності прогнозування

фінансових ринків зосереджуються на показниках точності, таких як середня абсолютна помилка (MAE) або коефіцієнт детермінації (R^2), але не враховують важливі аспекти ризику та адаптивності до змін на ринку. Врахування цих факторів є необхідним для оцінки стійкості та надійності моделі в умовах динамічного середовища.

Постановка задачі.

Завданням цього дослідження є розробка методики оцінки ефективності моделі прогнозування поведінки фінансових ринків, яка враховує як ризики, пов'язані з прогнозом, так і адаптивність моделі до змін ринкових умов. Введення таких показників дозволить підвищити надійність прийняття рішень у торговельних стратегіях та зробить модель більш адаптованою до реальних умов ринку.

Мета дослідження.

Розробити методику оцінки ефективності прогнозних моделей, яка враховує як точність прогнозу, так і його аспекти ризику та здатність адаптуватися до змін у структурі фінансового ринку. Мета полягає у створенні комплексного підходу до оцінки ефективності, що допоможе підвищити точність і стабільність прогнозів в умовах мінливого ринку.

Результати дослідження.

Для оцінювання ефективності моделей прогнозування фінансових ринків існують такі методики, як R^2 , MAE, MSE, Precision, Recall, F1-score. Кожна з них має свої недоліки, наприклад складність інтерпретації, великі похибки при динамічних змінах стану ринку, відсутність врахування ризиків тощо. Вони можуть бути складними для інтерпретації, демонструвати значні похибки при динамічних змінах ринку. Крім того, ці метрики часто показують різний рівень успішності залежно від ринкових умов, забезпечуючи якісну оцінку при стабільній поведінці фінансового інструменту, але даючи похибки під час різких коливань. Варто зауважити, що жоден з цих компонентів не використовує у своїх розрахунках компоненту ризиків, які були використані певною моделлю прогнозування та її адаптивності до зміни ситуації на

ринку. Дослідження показують, що здатність реагувати на різні ринкові умови та їхню динаміку, а також врахування потенційних ризиків є надзвичайно важливими критеріями оцінки ефективності моделі. Проте використання інструментів, які враховували б ці аспекти, вимагає великої кількості даних і обчислювальних потужностей, а також ускладнює інтерпретацію результатів роботи [1-2].

На основі вивчення характеристик існуючих методик, їхніх переваг та недолік, було запропоновано новий метод — **Адаптивний Коефіцієнт Ефективності (Adaptative Market Efficiency Score, AMES)**, який враховує:

Компоненту Прогнозування (PC), яка оцінює точність моделі (наприклад, через RMSE).

Компоненту Адаптивності (AC), яка визначає рівень адаптації моделі до змін волатильності або ринкових циклів на основі зміни значення моделі прогнозування, що аналізується. Наприклад, для Mean Square Error (MSE)

$$AC = 1 - \Delta RMSE_N$$

де

$N - \Delta RMSE$ в періоді "стандартного" стану ринку, що може бути пораховано як середня зміна помилки за певний період часу.

Формула зміни середньоквадратичної помилки протягом періоду:

$$\Delta RMSE = RMSE_t - RMSE_{t-1}$$

де

$RMSE_t$ — помилка в поточному періоді,

$RMSE_{t-1}$ — помилка в попередньому періоді.

Компоненту Ризику (RC), що включає оцінку ризику через метрику Value at Risk (VaR). Цей показник розраховується через вибірку значень історичних збитків, що не перевищують заданий рівень довіри:

$$RC = 1 - RR_{max}$$

де

R – VaR, максимальні збитки для заданого рівня довіри,

Rmax – максимальні можливі збитки.

Формула AMES виглядає наступним чином:

$$AMES = w_1PC + w_2AC - w_3RC$$

де

w_1, w_2, w_3 — ваги, що відповідають важливості кожної з компонент, і можуть бути налаштовані залежно від конкретних умов.

Запропонована нова методика AMES забезпечує більш комплексну оцінку ефективності моделі, ніж існуючі підходи, завдяки врахуванню не лише точності прогнозів, але і використанню в формуванні оцінці якості моделі прогнозування компоненти ризиків та адаптивності до змін становища на ринку.

Висновки та перспективи.

Розроблена методика AMES дозволяє покращити якість моделей прогнозування фінансових ринків, враховуючи як точність прогнозу, так і його здатність до адаптації та ризику. Це є особливо важливим для трейдерів та інвесторів, які прагнуть приймати рішення на основі комплексного аналізу.

Список використаних джерел:

1. Pomorsk P. Improving on the Markov-Switching Regression Model by the Use of an Adaptive Moving Average [Електронний ресурс] / P. Pomorsk, D. Gorse. – 2022. – Режим доступу до ресурсу: <https://arxiv.org/pdf/2208.11574>.
2. Chatigny P. Spatiotemporal adaptive neural network for long-term forecasting of financial time series [Електронний ресурс] / P. Chatigny, S. Wang // International Journal of Approximate Reasoning. – 2021. – Режим доступу до ресурсу: <https://www.sciencedirect.com/science/article/abs/pii/S0888613X2030267X>.

Ефективність CSRF-токенів у запобіганні міжсайтовим запитам у фронтенд-додатках

*Білодід Даніель Вадимович,
студент 6 курсу, групи ПДМ-63
Державного університету інформаційно-комунікаційних технологій
danielbeloded@gmail.com
Шахматов Іван Олександрович
Викладач кафедри ПЗ
Державного університету інформаційно-комунікаційних технологій
ivan.shakhmatov@gmail.com
Науковий керівник: Довженко Тимур Павлович,
доцент кафедри, к.т.н.
Державний університет інформаційно-комунікаційних технологій, м. Київ*

Постановка задачі.

Атаки типу Cross-Site Request Forgery (CSRF) становлять серйозну загрозу безпеці веб-додатків, оскільки дозволяють зловмисникам виконувати несанкціоновані дії від імені користувачів без їхнього відома. Це може призвести до компрометації конфіденційних даних, фінансових втрат та пошкодження репутації компаній. Питання ефективності використання CSRF-токенів у запобіганні таким атакам залишається актуальним, і потребує глибокого аналізу та дослідження для забезпечення надійного захисту фронтенд-додатків.

Мета дослідження.

Метою дослідження є оцінити ефективність CSRF-токенів як методу запобігання міжсайтовим запитам у фронтенд-додатках. Дослідження спрямоване на аналіз існуючих механізмів генерації, передачі та перевірки CSRF-токенів, а також на виявлення потенційних вразливостей і методів їх обходу зловмисниками. На основі отриманих даних будуть розроблені рекомендації щодо оптимального впровадження CSRF-токенів, що сприятиме підвищенню безпеки та надійності веб-додатків.

Практична реалізація

Для забезпечення захисту фронтенд-додатків від атак типу CSRF було обрано використання фреймворку Express.js, який є широко вживаним інструментом для

розробки серверних додатків на платформі Node.js. На початковому етапі встановлено npm-пакет Express для створення основи серверної частини додатка. Для інтеграції механізму захисту від CSRF-атак було додано пакет csurf, який спеціалізується на запобіганні міжсайтовим запитам, здатним спричинити несанкціоновані зміни в додатку.

Після встановлення пакету csurf його було інтегровано як middleware у серверний додаток на базі Express.js. Використання csurf як проміжного програмного забезпечення дозволяє автоматично генерувати CSRF-токени для кожного користувацького сеансу або запиту. Ці токени використовуються сервером для валідації вхідних запитів, що сприяє блокуванню неавторизованих дій та підвищує рівень безпеки додатка.

Усі форми та запити, які передбачають зміну даних на сервері, були модифіковані для включення CSRF-токенів. Сервер генерує токен і передає його клієнту, зазвичай через приховане поле форми або в заголовку відповіді. При наступному запиті клієнт зобов'язаний включити цей токен, що дозволяє серверу успішно провести верифікацію та підтвердити легітимність дії користувача.

При отриманні запиту сервер здійснює перевірку наявності та коректності CSRF-токена, переданого клієнтом. Це досягається шляхом порівняння токена з тим, що зберігається у сесії або cookie файлі. У разі відсутності токена або невідповідності значень сервер відхиляє запит і генерує відповідне повідомлення про помилку. Такий механізм забезпечує ефективний захист від несанкціонованих запитів, які можуть бути ініційовані зловмисниками.

CSRF-токени зазвичай зберігаються на сервері у сесії або передаються клієнту через cookie з атрибутом HttpOnly для підвищення безпеки. Збереження токенів у сесії дозволяє серверу мати контроль над їхнім життєвим циклом та забезпечує можливість їхньої перевірки при кожному запиті. Такий підхід мінімізує ризик компрометації токенів та знижує ймовірність успішного проведення CSRF-атаки.

Запропонована методика впровадження CSRF-токенів у серверний додаток на базі Express.js з використанням пакету csrf демонструє ефективний підхід до захисту від міжсайтових запитів. Інтеграція токенів у клієнтські запити та їхня перевірка на сервері забезпечує високий рівень безпеки та зменшує ризик несанкціонованих дій. Даний підхід може бути рекомендований для використання у сучасних веб-додатках, де безпека даних користувачів є пріоритетом.

Результати дослідження.

CSRF-токени є популярним механізмом безпеки завдяки використанню унікальних ідентифікаторів, згенерованих сервером та доданих до кожного запиту. Ці токени передаються через схему "запит-відповідь" і перевіряються сервером для автентифікації запиту, що дозволяє захистити користувача від несанкціонованих дій (Рис. 1).

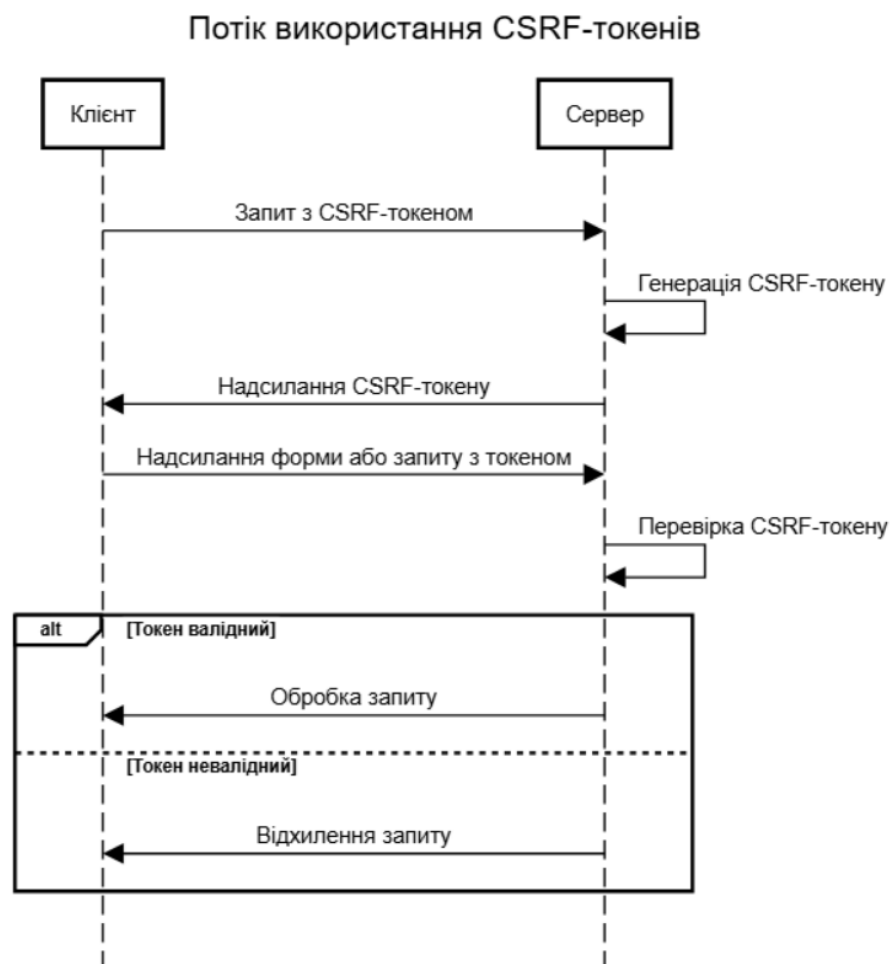


Рис 1. Потік використання CSRF-токенів для захисту від CSRF-атак у фронтенд-додатках.

Результати проведеного дослідження свідчать про те, що використання CSRF-токенів забезпечує ефективний захист користувачів від міжсайтових запитів завдяки їхній унікальності та складності підробки. Цей механізм надає надійний рівень безпеки, запобігаючи виконанню шкідливих запитів на сервері від імені користувача. Простота впровадження CSRF-токенів є значною перевагою, оскільки більшість серверних фреймворків підтримують автоматичну генерацію і перевірку таких токенів, що спрощує процес їх інтеграції в існуючі системи.

Крім того, CSRF-токени не вимагають від користувача додаткових дій, що робить їх використання повністю прозорим і не впливає на користувацький досвід. Проте слід враховувати певні недоліки та обмеження цього підходу. Зокрема, необхідність зберігання CSRF-токенів може спричинити проблеми з обробкою сеансів і вимагає ретельного захисту від можливого витоку або компрометації даних.

Висновки та перспективи.

Використання CSRF-токенів підтвердило свою ефективність як надійного засобу захисту веб-додатків від атак типу CSRF. Впровадження цього механізму дозволяє забезпечити високий рівень безпеки, особливо у випадках роботи з чутливими даними. Успішна інтеграція CSRF-токенів також сприяє покращенню досвіду користувача, оскільки їхнє функціонування відбувається у фоновому режимі без необхідності додаткових перевірок або дій з боку користувача.

Подальші дослідження в цьому напрямку можуть бути спрямовані на вдосконалення механізмів зберігання токенів, щоб мінімізувати ризики витоку або компрометації даних. Також перспективним є вивчення можливостей оптимізації використання CSRF-токенів у мультиплатформних додатках, що дозволить забезпечити узгоджений рівень безпеки в різних середовищах. Розробка нових методів генерації та перевірки CSRF-токенів може підвищити ефективність захисту та адаптувати механізм до сучасних вимог безпеки.

Список використаних джерел

1. Barth, A., Jackson, C., Mitchell, J. C. Robust Defenses for Cross-Site Request Forgery // Proceedings of the 15th ACM Conference on Computer and Communications Security. – 2008. – DOI: <https://doi.org/10.1145/1455770.1455782>.
2. OWASP. Cross-Site Request Forgery (CSRF). – OWASP Foundation, 2023. – [Електронний ресурс]. – Режим доступу: <https://owasp.org/www-community/attacks/csrf>.
3. Rana, A. K., Arora, S. A Review on Web Application Security: A Research Plan // Galgotias College of Engineering & Technology. – March 2018. – [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/338502196_A_Review_on_Web_Application_Security_A_research_plan.
4. Training Neural Networks by Enhance Grasshopper Optimization Algorithm for Spam Detection System / S. A. A. Ghaleb, et al. // IEEE Access. – 2021. – Vol. 9. – P. 116768–116813. – DOI: <https://doi.org/10.1109/access.2021.3105914>.
5. Agrawal, S. Mitigating Cross-Site Request Forgery (CSRF) Attacks Using Reinforcement Learning and Predictive Analytics // Applied Research in Artificial Intelligence and Cloud Computing. – 2023. – Vol. 6, No. 9. – P. 17–30. – [Електронний ресурс]. – Режим доступу: <https://www.researchberg.com/index.php/araic/article/view/189>.

Аналіз ефективності ux-оптимізації в ігрових проєктах на основі інформаційних систем аналітики поведінки користувачів

*Юхта Максим Артемович
Студент групи ПДМ-53
спеціальність 121 Інженерія програмного забезпечення
Державного університету інформаційно-комунікаційних технологій
BezNika2012@gmail.com
Науковий керівник: Компан Сергій Володимирович
кандидат фізико-математичних наук
доцент кафедри Інтернет-технологій
Державного університету інформаційно-комунікаційних технологій*

Ігрові веб-платформи є важливими точками контакту між гравцями та розробниками. Вони забезпечують доступ до інформації про гру, спільноти та додаткові послуги, такі як завантаження чи покупки. В умовах конкуренції в індустрії ігор стає критично важливим не лише створити естетичний дизайн, а й забезпечити зручність, швидкість та ефективність взаємодії з користувачами. Для цього застосовуються сучасні аналітичні системи, які допомагають вивчати поведінку гравців на платформі та впроваджувати зміни, що покращують їхній досвід.

Постановка задачі.

Актуальність оптимізації UX веб-ресурсів в ігрових проєктах обумовлена необхідністю підвищення залученості гравців, зменшення відсотка відмов і покращення конверсії. Задачею дослідження є визначення ключових метрик ефективності UX-оптимізації та впровадження аналітичних систем для аналізу поведінки користувачів, що дозволяє ідентифікувати проблемні зони в інтерфейсі і вдосконалювати взаємодію гравців із платформою.

Мета дослідження.

Метою є оцінка впливу UX-оптимізації на поведінку користувачів веб-ресурсів ігрових проєктів, використовуючи сучасні інструменти аналітики, такі як теплові карти, user journey mapping, та показники, включаючи середній час перебування на сторінці, коефіцієнт переходів (CTR) і відсоток відмов.

Результати дослідження.

Для оцінки ефективності UX-оптимізації використовується інтеграція інструментів Google Analytics, Hotjar і Pendo. Теплові карти, отримані за допомогою Hotjar, дозволяють визначити елементи, які користувачі активно ігнорують, що може свідчити про їхню нефункціональність або неправильне розташування.

User journey mapping допомагає побудувати маршрут користувача на платформі, від початкового входу до виконання ключової дії, наприклад, реєстрації чи покупки. Аналіз цих маршрутів дозволяє визначити, на якому етапі користувачі залишають платформу, що дає змогу впроваджувати точкові зміни.

Після оптимізації таких елементів, як навігаційне меню, сторінка входу та інформаційна архітектура, зафіксовано збільшення середнього часу перебування на сторінці на 18% і зменшення відсотка відмов із 45% до 32%.

Інформаційні системи дозволяють також відслідковувати технічні проблеми, наприклад, тривалість завантаження сторінок. Оптимізація швидкості завантаження, досягнута за допомогою впровадження кешування й адаптації зображень для різних пристроїв, скоротила середній час завантаження на 1,2 секунди, що позитивно вплинуло на конверсії.

Інтеграція динамічних рекомендацій на основі штучного інтелекту дозволила персоналізувати контент для кожного користувача. Це сприяло підвищенню коефіцієнта CTR на 24% завдяки пропозиціям ігор, які відповідають інтересам гравців, визначеним за даними про їхню активність.

Висновки та перспективи.

Використання інформаційних систем для аналітики поведінки користувачів показує високу ефективність в ідентифікації слабких місць у UX веб-ресурсів. Запропонований підхід дозволяє не лише підвищити основні метрики залучення, але й створити адаптивну систему, яка постійно вдосконалюється залежно від змін у поведінці користувачів.

Подальші дослідження можуть бути спрямовані на автоматизацію процесів UX-оптимізації шляхом використання алгоритмів машинного навчання, які будуть у реальному часі аналізувати поведінку користувачів і пропонувати відповідні зміни для покращення взаємодії з платформою.

Список використаних джерел:

1. UX Optimization - 4 Steps to Deliver a Better User Experience. Bluespace. – [Електронний ресурс] – Режим доступу: <https://uxcam.com/blog/user-experience-ux-optimization>.
2. User Interface Design and UX Design: 80+ Important Research Papers Covering Peer-Reviewed and Informal Studies. UX / UI Design and Professional Usability Testing | MAURO Usability Science. – [Електронний ресурс] – Режим доступу: <https://www.maurousabilityscience.com/blog/important-peer-reviewed-and-informally-published-recent-research-on-user-interface-design-and-user-experience-ux-design/>.
3. How behavioral analytics and UX work together. Capturly Blog. – [Електронний ресурс] – Режим доступу: <https://capturly.com/blog/how-behavioural-analytics-and-ux-work-together/>.
4. Gamification Concept in UX Design with Proven Examples (2025). Design Studio. – [Електронний ресурс] – Режим доступу: <https://www.designstudiouiux.com/blog/gamification-ux-design/>.

СЕКЦІЯ 4. СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ТА НЕЙРОННІ МЕРЕЖІ

Аналіз і прогнозування ризиків у стартапах за допомогою сучасних ML-алгоритмів

*Дзядевич Дмитро Дмитрович,
студент групи ПД-44,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
dimadz2002@gmail.com*

*Науковий керівник: Садовенко Володимир Сергійович,
кандидат фізико-математических наук, доцент
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Стартапи як драйвери інноваційного розвитку мають високу динамічність, але водночас стикаються з численними ризиками, які можуть негативно вплинути на їхній успіх. Серед найпоширеніших ризиків: фінансові труднощі, технологічні проблеми, невідповідність ринковим вимогам та конкурентний тиск.

Ефективний аналіз та прогнозування цих ризиків є критично важливим для забезпечення стабільності та зростання. У цьому контексті сучасні ML-алгоритми, які дозволяють працювати з великими обсягами даних і будувати прогнозні моделі, стають ключовим інструментом у ризик-менеджменті стартапів.

Постановка задачі.

Актуальним завданням є дослідження можливостей сучасних ML-алгоритмів для аналізу та прогнозування ризиків стартапів, що дозволяє покращити управлінські рішення і знизити рівень невизначеності.

Мета дослідження.

Вивчення ефективності ML-алгоритмів для прогнозування ризиків у стартапах та побудова інтегрованої моделі, яка дозволяє врахувати різні аспекти ризику.

Результати дослідження.

Для вирішення поставленої задачі були досліджені можливості ML-алгоритмів, таких як Decision Trees, Random Forest, Gradient Boosting та нейронні мережі.

Алгоритми дозволяють автоматизувати наступні процеси:

Виявлення ключових ризиків. Алгоритми кластеризації, такі як K-Means, групують схожі за характеристиками ризику.

Прогнозування фінансових загроз. Регресійні моделі оцінюють ймовірність виникнення касових розривів.

Аналіз поведінкових даних клієнтів. Нейронні мережі допомагають передбачити ризики відтоку клієнтів.

Етапи впровадження включають:

Формування бази даних. Збір внутрішніх та зовнішніх даних (фінансові показники, ринкові тенденції, поведінкові дані).

Обробка даних. Нормалізація, заповнення пропусків і формування навчального набору для моделювання.

Навчання моделей. Використання інструментів, таких як Scikit-learn та TensorFlow, для створення прогнозних моделей.

Оцінка точності моделей. Визначення ефективності за допомогою метрик (ROC-AUC, RMSE).

Практичний приклад: у стартапі з розробки програмного забезпечення використання Gradient Boosting дозволило ідентифікувати основні ризики, пов'язані з затримками у виконанні проєкту, що допомогло скоротити терміни реалізації на 15%.

Висновки та перспективи.

Сучасні ML-алгоритми є потужним інструментом для аналізу ризиків у стартапах, забезпечуючи точність, швидкість і автоматизацію процесів.

Подальші дослідження можуть бути спрямовані на адаптацію ML-алгоритмів до галузевих специфік, таких як фінанси, логістика чи e-commerce.

Список використаних джерел:

1. Савченко О., Прокопенко Є. "Методика оцінки ризиків", Збірник наукових праць, 2018.
2. Scikit-learn Documentation – [Електронний ресурс]. Режим доступу: <https://scikit-learn.org>.
3. Hinton G. "Deep Learning in Risk Management", Journal of AI, 2021.

Аналіз алгоритму «Зворотного поширення помилки» для навчання нейронних мереж»

*Кутняк Максим Юрійович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
maxim.kutnyak@gmail.com*

*Науковий керівник: Яскевич Владислав Олександрович,
к.т.н., доцент кафедри Інженерії програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій*

В наші часи нейронні мережі стали вже досить популярним інструментом який доступний майже кожному в кого є телефон, планшет, ноутбук, комп'ютер або навіть браслет з'являються з вбудованим штучним інтелектом. Але перед тим як користувач буде використовувати штучний інтелект його необхідно навчити, тому алгоритми навчання нейронних мереж є досить актуальною темою в наш час.

Постановка задачі.

Поставленою задачею є дослідження алгоритму «Зворотного поширення помилки» для навчання нейронних мереж.

Мета дослідження.

Метою даного дослідження є визначення слабких та сильних сторін алгоритму «Зворотного поширення помилки» навчання нейронних мереж.

Результат дослідження.

Спершу необхідно розібратись що ж таке алгоритм навчання нейронних мереж.

У спрощеному вигляді, алгоритм навчання – це набір правил, які дозволяють нейронній мережі змінювати свої внутрішні параметри (ваги) таким чином, щоб мінімізувати різницю між її прогнозами і реальними значеннями. Іншими словами, мережа намагається "підлаштуватися" під дані [1].

Зворотне поширення помилки – це один найпоширеніших алгоритмів, що використовується для навчання багат шарових перцептронів (MLP) та інших типів нейронних мереж [3].

Складається цей алгоритм з наступних кроків:

- Пряме поширення: на цьому кроці подаються вхідні дані на вхід мережі, після чого сигнал проходить через всі шари мережі, обчислюються значення нейронів у кожному шарі та на виході ми отримуємо прогноз;
- Обчислення помилки: на цьому кроці прогноз з попереднього кроку порівнюється з бажаним результатом та обчислюємо функцію втрат, яка кількісно оцінює різницю між прогнозом та бажаним результатом;
- Зворотне поширення: на цьому кроці помилка поширюється в зворотному напрямку, від останнього шару, до початку, на кожному шарі обчислюється градієнт функції втрат відносно ваг цього шару та ваги коригуються пропорційно градієнту, що мінімізує функцію втрат;
- Повторення: попередні кроки повторюються доти, доки помилка не стане прийнятною, або не буде досягнуто максимальної кількості епох навчання.

До переваг цього алгоритму відносять [2]:

- Цей алгоритм самостійно знаходить оптимальні значення ваг, мінімізуючи функцію втрат;

- Цей алгоритм може бути застосований до широкого спектру нейронних мереж, від простих багат шарових перцептронів до складних конволюційних і рекурентних мереж.
- Його можна застосовувати до різних типів нейронних мереж та задач;
- Зворотне поширення є основою для навчання глибоких нейронних мереж, які здатні вирішувати складні завдання, такі як розпізнавання зображень, обробка природної мови та генерація тексту.
- Існує багато способів оптимізації для прискорення процесу навчання.

Виокремлюють такі недоліки цього алгоритму:

- Функція втрат може мати багато локальних мінімумів. Алгоритм може застрягти в одному з них, не досягнувши глобального мінімуму через що модель може не навчитися оптимально виконувати завдання;
- У глибоких мережах градієнти можуть ставати дуже малими під час зворотного поширення, що призводить до повільного навчання або взагалі до зупинки навчання нижніх шарів через що мережа може не навчитися витягувати складні ознаки з даних;
- Градієнти можуть ставати дуже великими, що призводить до нестабільності навчання і розбіжності моделі через що можуть стати дуже великими, що ускладнює навчання і призводить до перенавантаження;
- Для великих мереж і великих наборів даних навчання може займати багато часу через що розробка моделі буде займати багато часу;
- Для навчання складних моделей потрібні великі і якісні набори даних які може бути складно отримати для рішення деяких задач.

Список використаних джерел:

1. Що таке зворотне поширення?. *Unite.AI*. URL: <https://www.unite.ai/uk/що-таке-зворотне-поширення/> (дата звернення: 14.10.2024).
2. Зворотне поширення в нейронній мережі: алгоритм машинного навчання. *Guru99*. URL: <https://www.guru99.com/uk/backpropogation-neural-network.html> (дата звернення: 24.11.2024).
3. GeeksforGeeks. Backpropagation in Neural Network - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/backpropagation-in-neural-network/> (дата звернення: 14.10.2024).

Використання нейронних мереж для автоматизації мастерингу аудіо

*Лабун Євген Миколайович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
kailas11252001@gmail.com
Науковий керівник: Золотухіна Оксана Анатоліївна,
кандидат технічних наук, доцент,
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Мастеринг аудіо — це складний процес, який залежить від висококваліфікованих звукорежисерів. З огляду на зростання обсягів музичного контенту зростає також потреба в обробці цього контенту. Використання нейронних мереж може значно спростити роботу звукорежисерів та автоматизувати багато аспектів мастерингу, зокрема корекцію тону, гучності, АЧХ [5] (амплітудно-частотної характеристики) та стереобалансу.

Постановка задачі.

Задачею дослідження є створення та навчання нейронної мережі для обробки аудіо файлів з метою покращення загальної якості їх звучання.

Мета дослідження.

Головна мета полягає в пошуку та оцінці найбільш ефективної та швидкодіючої моделі нейронної мережі [1], яка може виконувати мастеринг [5] аудіо треків різних жанрів і стилів без безпосередньої участі людини. Це включає визначення найкращих типів нейромереж, а також підходів та параметрів налаштування для тренування мережі, з метою досягнення оптимальної якості звуку.

Опис алгоритму роботи нейронної мережі для мастерингу аудіо

1. Підготовка даних:

- Збір даних: Зібрати велику кількість аудіотреків різних жанрів для тренування моделі.
- Аугментація даних: Використання бібліотеки Audiomentations для створення додаткових варіацій аудіо, щоб збільшити обсяг даних для тренування і зробити модель стійкішою до різних типів шумів і викривлень.

2. Попередня обробка аудіо:

- Перетворення аудіо: Перетворення сирого аудіо у спектрограму або інші форми представлення сигналу, які можуть бути подані на вхід нейронної мережі.
- Нормалізація: Нормалізація аудіосигналу для зменшення варіацій у гучності та інших параметрах.

3. Створення моделі:

- Архітектура моделі: Використання моделей CNN для виділення локальних ознак з спектрограм, моделей RNN для обробки послідовних даних та

моделей LSTM для збереження довготривалих залежностей у послідовностях аудіо.

- Об'єднання моделей: Комбінування різних типів нейромереж для досягнення найкращих результатів, використовуючи їхні сильні сторони.

4. Навчання моделі:

- Налаштування гіперпараметрів: Вибір оптимальних значень для гіперпараметрів моделі, таких як розмір фільтрів, кількість шарів, швидкість навчання тощо.
- Тренування моделі: Використання бібліотек TensorFlow або PyTorch для тренування моделі на підготовлених даних. Навчання моделі на великому обсязі аудіо даних для досягнення високої точності.
- Валідація: Перевірка якості моделі на валідаційному наборі даних та корекція параметрів у разі необхідності.

5. Тестування та оцінка:

- Тестування: Застосування натренованої моделі до нових аудіотреків для перевірки її здатності виконувати автоматичний мастеринг.
- Оцінка якості: Оцінка результатів за допомогою метрик якості звуку, таких як Signal-to-Noise Ratio (SNR), Total Harmonic Distortion (THD) та інших.

Результати дослідження.

Для розробки нейронної мережі використовуватиметься мова програмування Python та бібліотеки Audiomentations/Tensorflow[4]/PyTorch.

Очікується, що розроблена нейронна мережа зможе самостійно адаптуватися до варіацій в аудіо сигналах та виконувати мастеринг на рівні, порівнянному або кращому, ніж у випадку з ручним налаштуванням. Результати будуть демонструвати поліпшення в швидкості обробки треків та здатність системи ефективно працювати з великими обсягами даних [2][3].

Висновки та перспективи.

Застосування нейронних мереж у мастерингу аудіо відкриває нові можливості для музичних виконавців, звукорежисерів та продюсерів, зокрема зниження витрат і підвищення ефективності робочих процесів.

Універсальність та самостійність моделі нейронної мережі дозволяє інтегрувати її у професійні програми для роботи зі звуком та/або створення музики.

Список використаних джерел:

1. Nagesh B., Dr. M. Uttara Kumari – “A Review on Machine Learning for Audio Applications”, 05.2024. [Електронний ресурс] – Режим доступу: https://www.academia.edu/100503391/A_Review_on_Machine_Learning_for_Audio_Applications
2. Yi-Hsuan Yang – “Multimodal structure segmentation and analysis of music using audio and textual information”, 2009 IEEE International Symposium on Circuits and Systems, 05.2024. [Електронний ресурс] – Режим доступу: https://www.academia.edu/23085072/Multimodal_structure_segmentation_and_analysis_of_music_using_audio_and_textual_information.
3. Maria Navarro, Juan Manuel Corchado Rodriguez – “Machine Learning in Music Generation”, 05.2024. [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/326323583_Machine_Learning_in_Music_Generation
4. TensorFlow documentation, 05.2024. [Електронний ресурс] – Режим доступу: https://www.tensorflow.org/api_docs/python/tf/all_symbols
5. Michael Hewitt – “Music theory for computer musicians”, 05.2024. [Електронний ресурс] – Режим доступу: <https://toaz.info/doc-view-3>

Система обробки голосових повідомлень для гарячої лінії на основі методів машинного навчання

Левченко Олександр Олександрович,

студент групи ПДМ-62

спеціальність 121 Інженерія програмного забезпечення,

Державного університету інформаційно-комунікаційних технологій

mag6269128@stud.duikt.edu.ua

Шахматов Іван Олександрович

Викладач кафедри ІПЗ

Державного університету інформаційно-комунікаційних технологій

ivan.shakhmatov@gmail.com

Науковий керівник: Довженко Тимур Павлович,

доцент кафедри, кандидат технічних наук,

Державного університету інформаційно-комунікаційних технологій

Постановка задачі.

В наше сьогодення автоматизовані гарячі лінії набувають все більшого поширення в різних сферах діяльності, таких як обслуговування клієнтів, технічна підтримка, медичні консультації тощо. Одним із ключових аспектів їх ефективності є здатність швидко і точно обробляти голосові повідомлення від користувачів. Традиційні методи обробки голосових даних такі, як: шаблонне розпізнавання, статистичні методи та ін. Виявляються недостатньо ефективними через великий обсяг інформації, різноманітність мовленнєвих зразків та складність інтерпретації контексту. Розглянемо використання методів машинного навчання (ML) та їх комбінацію яка відкриває нові можливості для покращення та автоматизації цього процесу, забезпечуючи вищу точність та ефективність обробки голосових повідомлень.

Мета дослідження.

Метою дослідження є розробка та впровадження системи обробки голосових повідомлень для автоматизованої гарячої лінії на основі методів машинного навчання.

Це передбачає підвищення точності та ефективності обробки мовленнєвих даних шляхом подолання обмежень традиційних методів, таких як шаблонне розпізнавання та статистичні підходи.

Для досягнення поставленої мети необхідно проаналізувати існуючі методи та алгоритми обробки голосових повідомлень, вибрати та адаптувати найбільш підходящі методи машинного навчання для специфіки автоматизованих гарячих ліній, а також розробити прототип системи, що реалізує обрані алгоритми. Крім того, важливо оцінити ефективність розробленої системи в порівнянні з традиційними підходами, з урахуванням великого обсягу інформації, різноманітності мовленнєвих зразків та складності інтерпретації контексту. Це дослідження спрямоване на покращення якості та швидкості обробки голосових повідомлень, що в результаті підвищить ефективність автоматизованих гарячих ліній у різних сферах діяльності.

Результати дослідження.

У результаті проведеного дослідження здійснено детальний огляд літератури та аналіз існуючих методів обробки голосових повідомлень. На основі цього аналізу розроблено схему обробки голосових повідомлень (Рис.1). В якій вхідне голосове повідомлення надходить до системи. Далі це ж аудіо очищується від шуму та нормалізується. Потім за допомогою системи ASR (Automatic Speech Recognition) відбувається конвертування у текст. Виконується виділення ключових слів і базовий лінгвістичний аналіз. Текст класифікується для визначення типу запиту (наприклад, технічна підтримка, скарги тощо). І в кінці сформована відповідь записується у базу даних.

Було зібрано та попередньо оброблено голосові дані, що дозволило розробити та натренувати моделі машинного навчання, адаптовані до специфіки автоматизованих гарячих ліній.

Продуктивність розроблених моделей оцінювалася за допомогою метрик точності, F-міри, точності та повноти. Результати оцінювання показали підвищену ефективність моделей у порівнянні з традиційними підходами, особливо в умовах

великого обсягу інформації та різноманітності мовленнєвих зразків. Проведений аналіз отриманих результатів дозволив виявити сильні та слабкі сторони розробленої системи та сформулювати рекомендації щодо її подальшого вдосконалення.

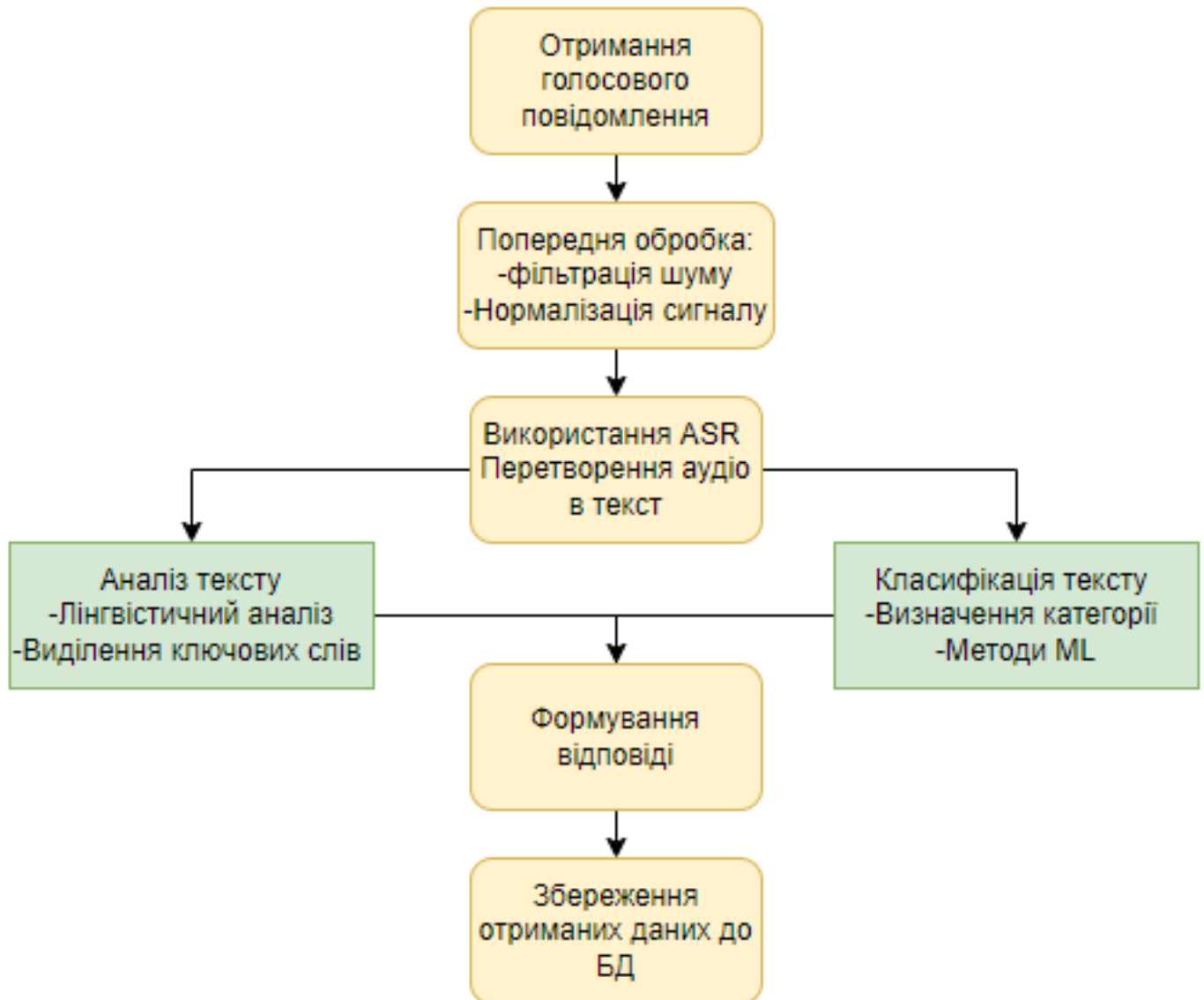


Рис. 1. Схема обробки голосових повідомлень

Висновки та перспективи.

Використання методики розпізнавання голосових повідомлень на основі методів машинного навчання значно покращить роботу автоматизованих систем гарячої лінії шляхом автоматичного перетворення голосових повідомлень у текст, точної класифікації запитів та швидкого реагування на потреби користувачів. Подальший розвиток може мати в собі розширення мовної підтримки, вдосконалення алгоритмів розпізнавання мови, інтеграцію з іншими системами та покращення безпеки даних. Це сприятиме подальшому розвитку автоматизованих гарячих ліній та підвищенню якості обслуговування.

Список використаних джерел:

1. Dessì D., Diego Reforgiato Recupero, Sack H. "An Assessment of Deep Learning Models and Word Embeddings for Toxicity Detection within Online Textual Comments". *MDPI*. [Електронний ресурс] – Режим доступу : <https://www.mdpi.com/2079-9292/10/7/779>
2. Svebeck R. Using machine learning to turn verbal conversations into structured contextual data records. [Електронний ресурс] – Режим доступу : <https://www.linkedin.com/pulse/using-machine-learning-turn-verbal-conversations-data-robert-svebeck>
3. M Tasbolatov, N. Mekebayev, O. Mamyrbayev, M. Turdalyuly, D. Oralbekova,. "Algorithms and architectures of speech recognition systems." *Psychology and Education Journal* 58, no. 2 (Лютий 20, 2021): 6497–501. [Електронний ресурс] – Режим доступу : <http://dx.doi.org/10.17762/pae.v58i2.3182>.
4. Ahmed, Ahmed M., and Aliaa K. Hassan. "Speaker Recognition Systems in the Last Decade – A Survey." *Engineering and Technology Journal* 39, no. 1B (Березень 25, 2021): 30–40. [Електронний ресурс] – Режим доступу : <http://dx.doi.org/10.30684/etj.v39i1b.1589>.

Визначення вимог до експертної системи для підтримки прийняття рішень на основі штучного інтелекту для лікарів первинної медицини

*Лукашенко Ігор Петрович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
egorkaluki46@gmail.com
Науковий керівник: Корецька В.О.,
к.пед.н., доцент, професор кафедри Інтернет-технологій
Державного університету інформаційно-комунікаційних технологій*

Сучасна охорона здоров'я приділяє значну увагу використанню інформаційних технологій для покращення догляду за пацієнтами та діагностики. Штучний інтелект (ШІ) демонструє великі перспективи в розробці експертних систем, які допомагають лікарям первинної ланки приймати обґрунтовані рішення. Ці системи здатні мінімізувати людські помилки та підвищити загальну якість надання медичної допомоги [1, 2, 3]. Однак створення таких систем потребує встановлення чітких критеріїв їхньої функціональності, точності та зручності для користувачів.

Постановка задачі.

Задачею дослідження є визначення ключових вимог до експертної системи на основі ШІ, яка забезпечить підтримку прийняття рішень лікарями первинної медицини.

Для цього необхідно:

Проаналізувати математичні методи і алгоритми класифікації медичних даних.

Дослідити сучасні інструменти програмної інженерії, що забезпечують створення таких систем.

Провести експериментальне тестування обраних підходів на прикладі наборів медичних даних.

Мета дослідження.

Метою дослідження є обґрунтування архітектури експертної системи та визначення ефективних методів і технологій розробки, які забезпечать високу точність, інтерпретованість результатів та зручність використання для медичних працівників.

Результати дослідження.

В ході дослідження було встановлено, що використання алгоритму Random Forest [4] для дослідження медичних даних, зокрема інформації про рак молочної залози, дозволило досягти точності класифікації на рівні 96%. Це підтверджується високими значеннями метрик точності, пригадування та F1-міри. Зокрема, точність для категорії «здорові» досягла 98%, а для категорії «хворі» - 96%.

Проведений аналіз важливості ознак дає можливість визначити, які фактори найбільш суттєво впливають на результат класифікації (рис. 2). Найвпливовіші ознаки включають найгіршу площу, найгірші точки увігнутості та середні точки увігнутості. Це підкреслює необхідність включення цих характеристик до моделі.

Для розробки системи та побудови моделі визначено сучасні програмні засоби інженерії програмного забезпечення. Рекомендовано використовувати бібліотеки Scikit-learn[5] та Matplotlib[6] для аналізу даних та візуалізації результатів.

Результати відображаються у вигляді матриці спутаності (рис. 1) і графіка важливості ознак (рис. 2).

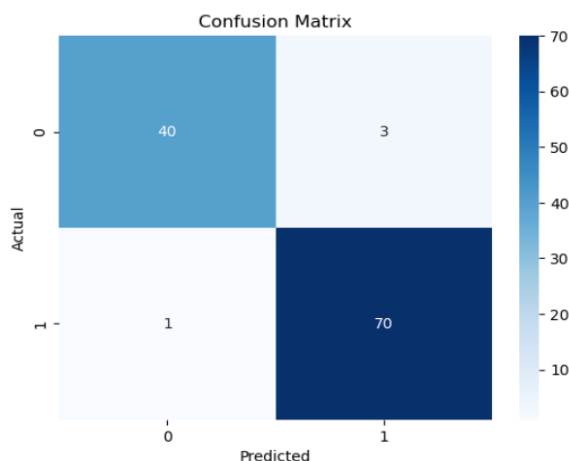


Рис. 1 Матриця спутаності моделі

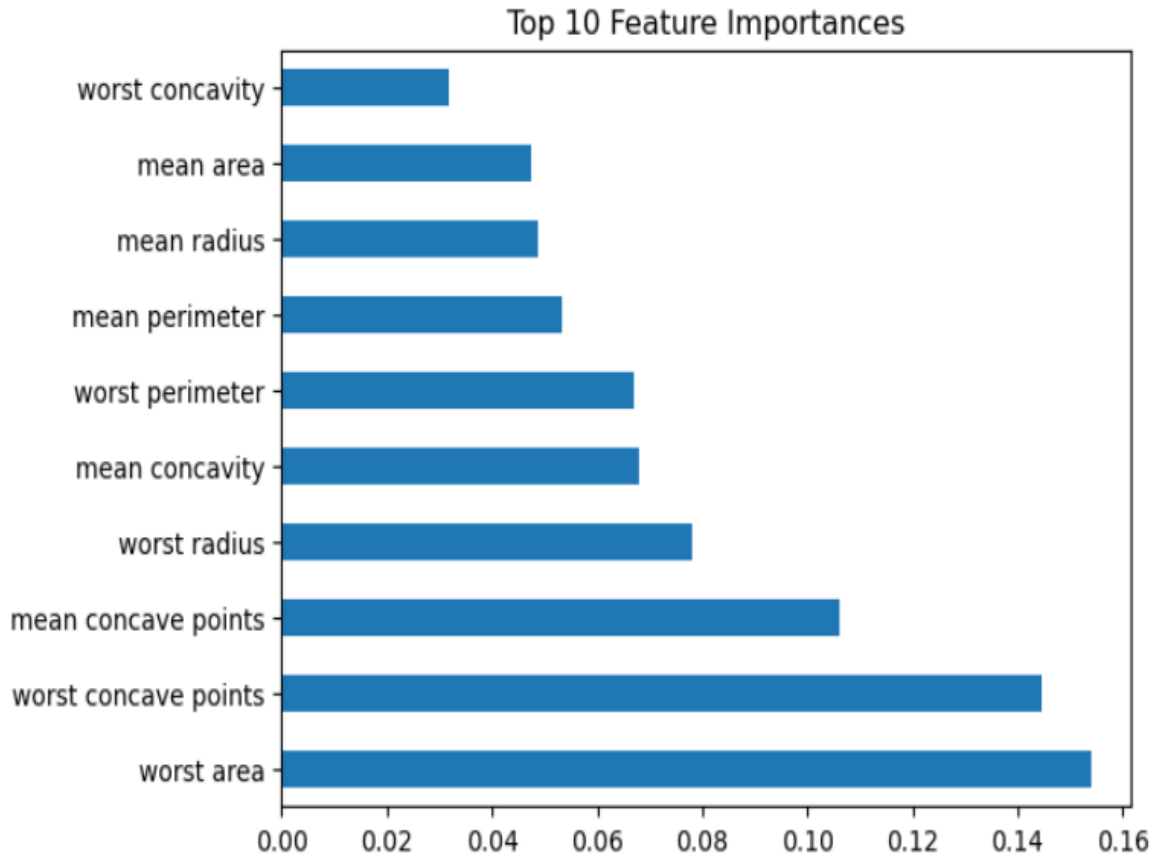


Рис. 2 Графік важливості ознак

Висновки та перспективи.

Дослідження показало, що використання методів машинного навчання, зокрема Random Forest, має потенціал для діагностичних цілей у закладах первинної медичної допомоги. Основні вимоги до системи включають високу точність, безперешкодну інтеграцію з сучасними медичними платформами та інтуїтивно зрозумілий користувацький інтерфейс. Подальші дослідження спрямовані на створення прототипу системи, здатної оцінювати з використанням реальних клінічних наборів даних.

Список використаних джерел:

1. An overview of clinical decision support systems: benefits, risks, and strategies for success - npj Digital Medicine / R. T. Sutton et al. Nature. URL: <https://www.nature.com/articles/s41746-020-0221-y> (date of access: 20.11.2024).
2. Esteva A. A guide to deep learning in healthcare - Nature Medicine. Nature. URL: <https://www.nature.com/articles/s41591-018-0316-z> (date of access: 20.11.2024).
3. Hastie T., Tibshirani R., Friedman J. Overview of supervised learning. SpringerLink. URL: https://link.springer.com/chapter/10.1007/978-0-387-84858-7_2 (date of access: 20.11.2024).
4. Random forest algorithms in health care sectors: a review of applications / G. Haripriya et al. Researchgate. URL: https://www.researchgate.net/publication/358128515_Random_Forest_Algorithms_in_Health_Care_Sectors_A_Review_of_Applications (date of access: 20.11.2024).
5. User guide. scikit-learn. URL: https://scikit-learn.org/stable/user_guide.html (date of access: 20.11.2024).
6. Using Matplotlib – Matplotlib 3.9.2 documentation. Matplotlib – Visualization with Python. URL: <https://matplotlib.org/stable/users/index> (date of access: 20.11.2024).

Business Logic Generation using AI technology

*Mykolaienko Vladyslav Oleksandrovich,
student of SE-32,
specialty 121 Software engineering,
State University of Information and Communication Technologies
mykolaienkovlad@gmail.com
Academic supervisor: Storchak Kamila Pavlivna,
Doctor of technical sciences, professor,
Head of the Department of Information Systems and Technologies
State University of Information and Communication Technologies*

We live in the AI era, where almost every electronic device can either be used for AI or can incorporate AI capabilities. This reality gives us the opportunity to automate a lot of hard work that, just a few years ago, could only be done by people. In this thesis, I aim to explore and provide examples of how we can automate business logic processes, as well as how to generate and structure them effectively.

Problem Statement.

AI technology has rapidly advanced, yet many businesses still handle complex tasks manually, leading to inefficiencies and errors. There is a need for effective ways to automate business logic using AI to improve accuracy and reduce costs. This thesis explores how AI can automate these processes, making them more efficient and adaptable.

Research Objective.

The objective of this research is to explore how AI can be used to automate and optimize business logic processes for greater efficiency and accuracy.

Research Results.

The result of this research is the development of an algorithm designed to assist developers and businesses in integrating AI into the generation of business logic. This algorithm streamlines the creation of business logic by automating complex decision-making processes, reducing manual effort, and minimizing human error. It provides a structured framework that enables organizations to adopt AI-driven solutions more effectively, allowing for scalable and efficient automation of routine business tasks. The algorithm is

specifically tailored to support both technical and non-technical users, facilitating a smoother transition to AI-enhanced business operations.

To reproduce my algorithm for business logic generation, you need to:

1. Install any available Large Language Model, such as "Llama 3.1".

2. Scan your project's code and configuration files, and create a `context_config` file.

This file should contain a high-level overview of your application, including the app modules, their basic structure, entities, and functions. This is how it looks like:

```
core_idea: "{{ core_idea }}" # This is the primary concept or purpose of the project (e.g., "T
entities:
- "{{ EntityName }}":
  structure:
    "{{ attribute1_name }}": "{{ attribute1_type }}" # Replace with attribute name and ty
    "{{ attribute2_name }}": "{{ attribute2_type }}"
  relationships:
    - "{{ RelatedEntityName1 }}" # List of related entities
    - "{{ RelatedEntityName2 }}"

- "{{ AnotherEntityName }}":
  structure:
    "{{ another_attribute1_name }}": "{{ another_attribute1_type }}"
    "{{ another_attribute2_name }}": "{{ another_attribute2_type }}"
  relationships:
    - "{{ RelatedEntityName3 }}"

business_functions:
- function_name: "{{ function_name1 }}" # Replace with the business function name
  input_params:
    - param_name: "{{ input_param1_name }}" # Replace with parameter name
      param_type: "{{ input_param1_type }}" # Replace with parameter type
    - param_name: "{{ input_param2_name }}"
      param_type: "{{ input_param2_type }}"
  output_params:
    - param_name: "{{ output_param1_name }}"
      param_type: "{{ output_param1_type }}"
  reason: "{{ business_function_reason1 }}" # Reason for the business function
```

Picture 1

3.Receive prompt from the User , tokenize it and pass to the Large Language Model

4.Create Generation Receipt using Large Language Model

```
templates > ! generation_config_recipe.yaml
1  Generation_Recipe:
2    - task: "{{ task_description_1 }}" # Task for the LLM to perform
3      filename: "{{ filename_full_project_path_1 }}" # Full project path
4      generation_task_context: "{{ generation_task_context_1 }}" # Context for the task
5      generation_task_output: "" # This field should remain empty
6
7    - task: "{{ task_description_2 }}" # Another task for the LLM
8      filename: "{{ filename_full_project_path_2 }}" # Full project path
9      generation_task_context: "{{ generation_task_context_2 }}" # Context for the task
10     generation_task_output: "" # This field should remain empty
11
12    - task: "{{ task_description_3 }}" # Additional tasks can be added here
13      filename: "{{ filename_full_project_path_3 }}"
14      generation_task_context: "{{ generation_task_context_3 }}"
15      generation_task_output: "" # This field should remain empty
16
```

Picture 2

This step is very important to understand , especially generation_task_context property, generation_task_context - it's a combined text from main context_config file LLM analyzes it , and takes only important context in order to perform operation.

5.Generate code from generation receipt using LLM

6.Test it

7.Build project

8.Add new generated content into config

Also I want to mention that this can be the universal approach for different tasks of generation.

Conclusions and Future Prospects.

The development of an AI-driven algorithm for automating business logic demonstrates the potential for significantly enhancing operational efficiency and reducing errors in business processes. This research highlights how AI can streamline complex workflows, allowing businesses to focus on higher-level decision-making and strategic planning. The algorithm's adaptability makes it suitable for a wide range of industries, showcasing its versatility and practical application.

Looking forward, there are several opportunities to expand upon this research. Future work could involve refining the algorithm to handle more specialized and industry-specific tasks, enhancing its ability to learn and adapt to unique business environments. Additionally, incorporating advanced AI techniques, such as natural language processing and machine learning, could enable the algorithm to better interpret unstructured data and user inputs. There is also potential for integrating the algorithm into existing business platforms, making AI-driven automation more accessible and user-friendly for a broader audience. Ultimately, this research opens new pathways for businesses to harness AI technology for optimizing and transforming their operations.

References:

1. M. Goodfellow, Y. Bengio, and A. Courville, **Deep Learning**, MIT Press, 2016, pp. 123-145.
2. K. Schwab, "The Fourth Industrial Revolution," **World Economic Forum**, 2017, pp. 78-92.
3. J. Pearl, **Causality: Models, Reasoning, and Inference**, 2nd ed., Cambridge University Press, 2009, pp. 230-255.
4. P. Domingos, **The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World**, Basic Books, 2015, pp. 101-115.

Адаптивне оновлення моделей розпізнавання дорожніх знаків на мобільних пристроях

*Пальчук Михайло Едуардович,
студент групи АППЗ-31,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
mykhailo.pal@gmail.com
Науковий керівник: Корецька Вікторія Олександрівна, к.пед.н.,
професор кафедри Інтернет-технологій
Державного університету інформаційно-комунікаційних технологій*

Системи розпізнавання дорожніх знаків для мобільних пристроїв стикаються з викликами, пов'язаними зі зміною умов використання: з'являються нові дорожні знаки, змінюються правила дорожнього руху, встановлюються тимчасові знаки. Адаптивне оновлення моделей без необхідності передачі великих обсягів даних на центральний сервер стає серйозним викликом для системи. У цій доповіді розглядаються методи адаптивного оновлення моделей машинного навчання, які дозволяють підвищити точність та актуальність системи, зберігаючи конфіденційність даних користувачів.

Постановка задачі.

Метою цієї статті є систематизація та аналіз останніх досліджень у сфері адаптивного оновлення моделей розпізнавання дорожніх знаків на мобільних пристроях. Зокрема, мета включає вирішення проблем доступу до вихідного датасету, на якому початкова модель навчалася, раціонального використання обмежених ресурсів мобільних пристроїв, збереження приватності даних користувача та забезпечення сумісності початкової та оновленої моделей.

Результати дослідження.

За допомогою трансферного машинного навчання [1] локальна модель покращується, продовжуючи навчання на невеликих обсягах даних, зібраних локально, без втрати приватності користувача.

Оновлена модель без прив'язки до вихідних даних передається на сервер, де основна модель вдосконалюється за допомогою федеративного навчання[2][3]. Використання фреймворку CoreML [4], який оптимізований компанією Apple для мобільної системи iOS, забезпечує ефективне оновлення і навчання локальної моделі у фоновому режимі, без надмірного використання ресурсів і без впливу на продуктивність інтерфейсу додатку. Даний підхід дозволяє підвищити точність розпізнавання нових знаків на 5–10% у порівнянні зі статичними моделями. Система демонструє стабільну роботу на мобільних пристроях у реальних умовах використання.

Висновки та перспективи.

Адаптивне оновлення моделей розпізнавання дорожніх знаків на мобільних пристроях є перспективним напрямом, що дозволяє забезпечити актуальність і точність системи в динамічно змінних умовах. Використання підходів, таких як трансферне і федеративне навчання, дозволяє вирішити проблеми приватності та обмеженості ресурсів, забезпечуючи при цьому високу продуктивність. Це відкриває нові можливості для інтеграції таких систем у широкий спектр мобільних додатків і пристроїв.

Список використаних джерел:

1. A Review of Deep Transfer Learning and Recent Advancements. URL: <https://www.mdpi.com/2227-7080/11/2/40> (дата звернення: 23.11.2024)
2. FedPE: Adaptive Model Pruning-Expanding for Federated Learning on Mobile Devices. URL: <https://ieeexplore.ieee.org/document/10463172> (дата звернення: 23.11.2024)
3. AdaptiveFL: Adaptive Heterogeneous Federated Learning for Resource-Constrained AIoT Systems. URL: <https://ar5iv.org/html/2311.13166> (дата звернення: 23.11.2024).
4. Personalizing a Model with On-Device Updates. URL: <https://developer.apple.com/documentation/coreml/personalizing-a-model-with-on-device-updates> (дата звернення: 23.11.2024).

Оптимізація процесу автоматизованого тестування за допомогою впровадження технології GitHub copilot

*Петрунчак Анна Романівна,
студентка групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
annpboyko@gmail.com*

*Шахматов Іван Олександрович
Викладач кафедри ПЗ
Державного університету інформаційно-комунікаційних технологій
ivan.shakhmatov@gmail.com*

*Науковий керівник: Довженко Тимур Павлович,
к.т.н., доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Процес автоматизованого тестування є важливою складовою частиною розробки програмного забезпечення, що дозволяє знижувати час тестування та підвищувати якість продуктів. Однак традиційні методи розробки тестів можуть бути часозатратними та потребують значних ресурсів. Впровадження технології GitHub Copilot для автоматизованого створення тестів може значно покращити цей процес, скорочуючи час на розробку тестових сценаріїв і збільшуючи ефективність тестування [1-2].

Мета дослідження.

Метою дослідження є аналіз та оцінка можливостей впровадження технології GitHub Copilot у процес автоматизованого тестування для підвищення ефективності розробки тестових сценаріїв та зменшення часу, необхідного для тестування програмного забезпечення. Це передбачає детальне вивчення впливу GitHub Copilot на процес створення автоматизованих тестів, оцінку його ефективності в порівнянні з традиційними методами, а також ідентифікацію потенційних переваг та обмежень при використанні цієї технології в реальних умовах розробки програмного забезпечення.

Предмет дослідження.

Предметом дослідження є процес автоматизованого тестування програмного забезпечення з впровадженням технології GitHub Copilot. Це включає в себе вивчення методів інтеграції GitHub Copilot у існуючі процеси тестування, аналіз його впливу на ефективність розробки тестових сценаріїв, а також оцінку змін у якості та швидкості тестування програмного забезпечення при використанні цієї технології

Постановка задачі.

Задачею дослідження є вивчення технології GitHub Copilot як інструмента для автоматизованого створення тестів, розробка методології інтеграції цієї технології у процес автоматизованого тестування та оцінка ефективності цього підходу.

Результати дослідження.

У результаті дослідження було розроблено методику використання GitHub Copilot для автоматизації процесу створення тестів. Застосування цієї технології дозволяє значно зменшити час, що витрачається на написання тестових сценаріїв, а також знизити ймовірність помилок у тестах за рахунок використання AI-підходів. Інтеграція Copilot в існуючі інструменти автоматизації тестування, такі як Selenium та JUnit, дає змогу прискорити процес тестування, не знижуючи його ефективності [2-4].

Висновки та перспективи.

Впровадження GitHub Copilot у процес автоматизованого тестування дозволяє скоротити час на створення тестових сценаріїв і підвищити точність тестування завдяки використанню штучного інтелекту. У майбутньому можна розглядати можливість застосування цієї технології для генерації більш складних тестових наборів, а також інтеграції з іншими інструментами для забезпечення високої якості тестування на всіх етапах розробки програмного забезпечення.

Список використаних джерел:

1. “Revolutionizing Automation Testing Efficiency with GitHub Co-Pilot” 2024. – [Електронний ресурс] – Режим доступу: <https://medium.com/@kunalnegi0002/revolutionizing-automation-testing-efficiency-with-github-co-pilot-53d1d24a4613>
2. “Passing the GitHub Copilot Certification: A Study Guide” 2024. – [Електронний ресурс] – Режим доступу: https://docs.github.com/en/enterprise-cloud%40latest/copilot/using-github-copilot/guides-on-using-github-copilot/writing-tests-with-github-copilot?utm_source=chatgpt.com
3. “Writing tests with GitHub Copilot” 2024. – [Електронний ресурс] – Режим доступу: https://docs.github.com/en/enterprise-cloud%40latest/copilot/using-github-copilot/guides-on-using-github-copilot/writing-tests-with-github-copilot?utm_source=chatgpt.com
4. “Leveraging GitHub Copilot for Testing: Revolutionizing Software Quality Assurance”, 2024. – [Електронний ресурс] – Режим доступу: <https://rahulqalabs.medium.com/leveraging-github-copilot-for-testing-revolutionizing-software-quality-assurance-3be07cc151e4>

Використання **Semantic Kernel** для оркестрації функціоналу **LLM** в розробці ПЗ

*Решетнік Нікіта Олександрович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
reshetnik.nikita@gmail.com
Науковий керівник: Трінтіна Наталія Альбертівна,
к.т.н., доцент, доцент кафедри Інтернет-технологій
Державного університету інформаційно-комунікаційних технологій*

У сучасному світі великі мовні моделі (LLM) стають невід'ємною частиною розробки інтелектуальних систем та додатків. Інтеграція таких моделей у програмне забезпечення може бути складним завданням через їхню складність та вимоги до

ресурсів. Semantic Kernel від Microsoft пропонує рішення для спрощення цього процесу, надаючи розробникам інструменти для ефективної оркестрації LLM у їхніх проектах.

Постановка задачі.

Задачею дослідження є вивчення використання Semantic Kernel для оркестрації функціоналу LLM у процесі розробки програмного забезпечення.

Мета дослідження.

Дослідити можливості та переваги використання Semantic Kernel при інтеграції LLM у програмні додатки, а також визначити потенційні виклики та шляхи їх подолання.

Результати дослідження.

Semantic Kernel — це набір інструментів та SDK від Microsoft, призначений для спрощення інтеграції LLM у додатки різного рівня складності. Він надає можливість дуже швидко інтегруватись до широкого спектру LLM, та використовувати їх в кодї без необхідності написання конекторів до API цих моделей. Більш того, в Semantic Kernel реалізовані механізми функцій та плагінів які можуть бути використані LLM для видачі актуальної відповіді.

Функції – це звичайний програмний код який приймає на вході якісь дані, ти віддає їх. Але важливою деталлю є те, що цей код можна декорувати відповідним атрибутом, і він стає доступний LLM для використання. Наприклад: уявимо будь-який онлайн магазин, в якого є інтелектуальний чат бот. За допомогою функцій ми зможемо давати чат боту той контекст, який потребується в запиті користувача. Наприклад користувач бажає дізнатись свої не оплачені замовлення. Він так і питає це в чат-бота. Чат бот в свою чергу виходячи з контексту, розуміє що йому потрібні замовлення користувача, викликає функцію запиту на сервер з відповідними даними, і використовує ці дані для обробки запиту користувача.

Плагіни – це модуль який в собі зберігає відповідні до його призначення функції. Наприклад один з примітивних вбудованих плагінів це є плагін часу. На

скільки ми знаємо, мовні моделі не мають доступу до реального часу, але за допомогою цього плагіну, ми даємо моделі такі можливості. В разі того якщо модель розуміє що їй потрібно зробити якісь операції з часом, чи знати поточний час для надання релевантної відповіді, вона просто викликає відповідну функцію в плагіні.

Переваги використання Semantic Kernel:

1. **Швидка інтеграція LLM у існуючі додатки Готове рішення:** більше не треба створювати свій бойлерплейт тільки для формування запитів в OpenAI тощо, для цього є готовий інструмент що підтримується великою компанією.
2. **Модульність та повторне використання коду:** просто дайте доступ до наявних в проекті функцій для LLM для виконання відповідних задач, і ви вже можете кратно розширити можливості вашої AI системи. І все це без необхідності писати окремий код для цього. Важлива нотатка це те що все ще не потрібно забувати про безпеку та не відправляти конфіденційні дані до моделі.
3. **Широкий вбудований функціонал:** Вже з «коробки» цей фреймворк дозволяє використовувати функціонал чату, переводу голосу в текст, та навпаки, генерацію зображень. Просто напишіть промпт та викликайте відповідний сервіс!
4. **Підтримка різних платформ та мов програмування.** Semantic Kernel підтримує інтеграцію з популярними мовами програмування такими як: .NET, Java, Python, що робить його універсальним інструментом.
5. **Швидка інтеграція LLM:** швидка інтеграція в існуючі додатки без потреби глибокого розуміння внутрішньої роботи моделей.

Мінуси використання:

- **Не очевидні витрати:** при автоматичному виклику функцій LLM часто може додавати собі не потрібний контекст, що може збільшити середнє використання токенів на запит.

- **Новизна фреймворку:** оскільки фреймворк можна вважати молодим (2024 рік), в ньому все ще присутня деяка пустота та відпрацьованість функціоналу. Щось працює в експериментальному режимі, щось потребує не зручного налаштування.
- **Мало документації та відпрацьованих практик:** це те що виплаває з минулого пункту, оскільки продукт ще молодий, ком'юніті ще не сформувало загально прийняті методики як робити те або інше. Деколи навіть на самі прості питання можна дуже довго шукти відповіді в коментарях на GitHub

Висновки та перспективи.

Використання Semantic Kernel для оркестрації функціоналу LLM у розробці програмного забезпечення відкриває широкі можливості для створення більш інтелектуальних та адаптивних додатків. Це дозволяє розробникам швидко інтегрувати передові технології без значних витрат часу та ресурсів. Подальше дослідження в цій області може призвести до появи нових методів оптимізації та безпечної інтеграції LLM у різні сфери застосування.

Список використаних джерел:

1. Microsoft Semantic Kernel Documentation – [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/semantic-kernel/>
2. GitHub - microsoft/semantic-kernel – [Електронний ресурс] – Режим доступу: <https://github.com/microsoft/semantic-kernel>
- Orchestrating Large Language Models with Semantic Kernel – [Електронний ресурс] – Режим доступу: <https://devblogs.microsoft.com/ai/orchestrating-llms-with-semantic-kernel/>
3. Integrating AI into Applications using Semantic Kernel – [Електронний ресурс] – Режим доступу: <https://www.infoq.com/articles/semantic-kernel-ai-integration/>

Розробка системи машинного навчання для оптимізації банківських операцій

*Соломоник Олексій Павлович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
alex.solomonik2000@gmail.com*

*Науковий керівник: Компан Сергій Володимирович,
кандидат фізико-математичних наук,
доцент кафедри Інтернет-технологій
Державного університету інформаційно-комунікаційних технологій*

Сучасний банківський сектор стикається з необхідністю автоматизації рутинних операцій, підвищення ефективності управління ризиками та персоналізації взаємодії з клієнтами. Машинне навчання є ключовим інструментом для вирішення цих завдань завдяки можливості обробки великих обсягів даних і створенню прогнозних моделей.

Постановка задачі.

Задачею дослідження є визначення ключових вимог до системи машинного навчання для впровадження в банківські операції. Метою роботи є розробка прототипу системи, яка дозволяє оптимізувати управління клієнтським портфелем, знизити фінансові ризики та автоматизувати виявлення шахрайства.

Мета дослідження.

Розробити прототип системи машинного навчання для прогнозування поведінки клієнтів та оптимізації банківських операцій з використанням сучасних алгоритмів і архітектурних підходів.

Результати дослідження.

Для розробки прототипу використовуються сучасні бібліотеки машинного навчання, зокрема Scikit-learn, TensorFlow та XGBoost.

Система базується на архітектурі "клієнт-сервер", де:

- Клієнтська частина: взаємодія через веб-інтерфейс для представлення результатів прогнозів.

- Серверна частина: обробка даних, навчання моделей та виконання прогнозів.

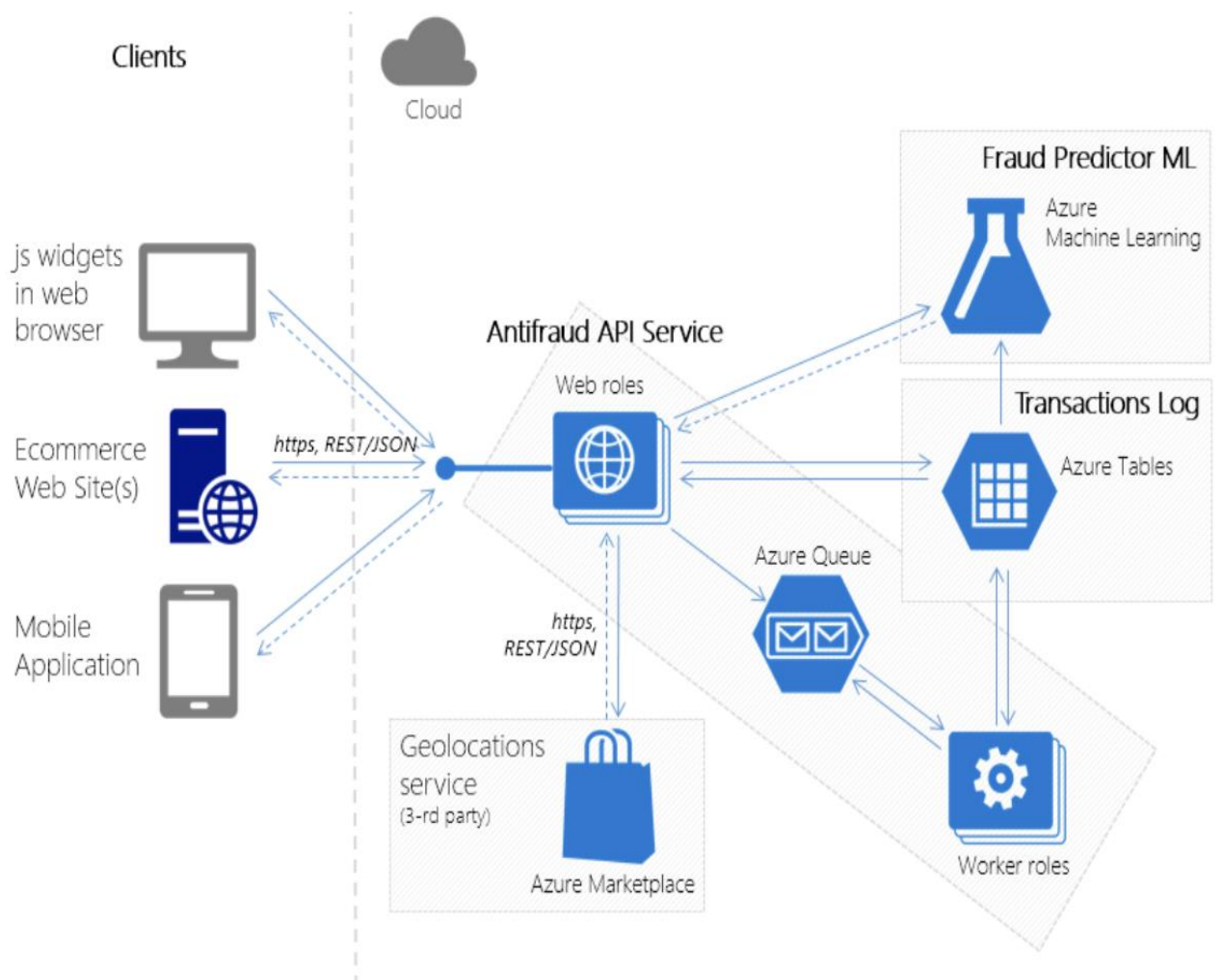


Рис. 1. Схема роботи системи

Ключові функціональні модулі системи:

- Прогнозування ризиків: використання класифікаційних моделей (Random Forest, XGBoost) для оцінки кредитоспроможності клієнтів.
- Антифрод-система: автоматичне виявлення підозрілих транзакцій на основі аномалій у поведінці клієнтів.

- Оптимізація клієнтських пропозицій: персоналізація банківських послуг з використанням моделей кластеризації (K-means).

- Прототип протестований на наборі даних Credit Card Fraud Detection Dataset та реальних транзакціях банківських клієнтів (знеособлені дані). Моделі показали точність прогнозів на рівні 85-95% залежно від задачі.

Висновки та перспективи.

Розроблена система дозволяє підвищити ефективність банківських операцій за рахунок автоматизації рутинних процесів, зменшення ризиків та підвищення якості обслуговування клієнтів. Подальший розвиток передбачає інтеграцію системи з існуючими банківськими платформами та підключення додаткових джерел даних для підвищення точності прогнозів.

Список використаних джерел:

1. G. Wang, J. Ma, L. Wang, "Credit Card Fraud Detection Using Machine Learning Algorithms," Journal of Financial Analytics, vol. 23, no. 4, pp. 45-57, 2021.
2. TensorFlow Framework Documentation – [Електронний ресурс]. – Режим доступу: [<https://www.tensorflow.org>].
3. Scikit-learn: Machine Learning in Python – [Електронний ресурс]. – Режим доступу: [<https://scikit-learn.org/stable/>].
4. XGBoost Documentation – [Електронний ресурс]. – Режим доступу: [<https://xgboost.readthedocs.io>].
5. "Credit Card Fraud Detection Dataset," Kaggle – [Електронний ресурс]. – Режим доступу: [<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>].

Використання моделі на основі напівконтрольованого машинного навчання для визначення пріоритетності пацієнтів лікарні

*Спіцин Андрій Ярославович,
студент 6 курсу, групи ПДМ-61*

*Державного університету інформаційно-комунікаційних технологій
and.spitsyn@gmail.com*

*Науковий керівник: Садовенко Володимир Сергійович,
кандидат фізико-математичних наук,
доцент, професор кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Сучасні відділення невідкладної допомоги щодня стикаються із великим потоком пацієнтів, серед яких необхідно швидко визначати найкритичніші випадки. Для оптимізації цього процесу важливо впроваджувати автоматизовані підходи, які базуються на аналізі медичних даних. Для створення адаптивної, точної та ефективної моделі пропонується використати методи напівконтрольованого машинного навчання [1-2]. У цьому тексті описано основні етапи створення такої моделі та пояснено, чому саме цей підхід є найбільш доцільним для поставленої задачі [3].

Розробку моделі можна розділити за декількома етапами:

Підготовчий етап у розробці будь-якої моделі машинного навчання полягає у зборі даних, необхідних для роботи моделі. У даному випадку йдеться про медичні дані, які включають інформацію про фізіологічні показники, демографічні характеристики, обставини надходження до лікарні, попередні діагнози та інші фактори, за якими можна визначити пріоритетність пацієнта. Після збору ці дані необхідно очистити та стандартизувати, адже підготовлені дані дозволяють моделі працювати стабільно та з мінімальними помилками.

Маючи підготовлені дані, можна переходити до першого етапу розробки. На першому етапі модель тренується на невеликому обсязі даних, де вже вказані правильні оцінки пріоритетності пацієнтів. Це дозволяє моделі зрозуміти основні зв'язки між характеристиками пацієнтів і необхідною терміновістю допомоги. Завдяки цьому модель отримує базу для подальшого удосконалення.

На другому етапі модель застосовує отримані знання до немаркованих даних, прогножуючи оцінки пріоритетності для пацієнтів. Дані, у яких модель найбільш упевнена, додаються до основного маркованого набору, і модель навчається повторно. Такий підхід дозволяє поступово розширювати базу маркованих даних без значних витрат, одночасно підвищуючи точність прогнозів. Саме тому цей фактор є одним із вирішальних у виборі методу машинного навчання. Завдяки ньому можна натренувати точну та адаптивну модель навіть не маючи значної бази маркованих даних.

Після цього можна переходити до тестування моделі. Моделі зазвичай перевіряються на тестових вибірках, які не використовувалися раніше. Вимірюються точність, чутливість, специфічність та інші метрики. Це допомагає оцінити, наскільки добре модель справляється з реальними задачами, та визначити можливі напрямки вдосконалення.

Кінцевий етап — впровадження моделі в медичну практику. Якщо тестування показують гарні результати, можна запропонувати впровадити модель у медичних закладах для більш реалістичного тестування на практиці. Оскільки систему можна налаштувати так, що модель отримує вхідні дані безпосередньо від медичної техніки, таке тестування не буде мати негативний вплив на роботу медичного персоналу.

Висновок

Розробка моделі на основі напівконтрольованого машинного навчання є раціональним і перспективним підходом до вирішення задачі оцінки пріоритетності пацієнтів. Вибраний метод дозволяє працювати з обмеженими ресурсами, поступово вдосконалюючи модель у процесі її використання. Завдяки поєднанню точності, гнучкості та інтеграції з реальними медичними системами, цей підхід має всі шанси стати ефективним інструментом для відділень невідкладної допомоги.

Список використаних джерел:

1. Study On Machine Learning Algorithms / P. R та ін. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2021. С. 67–72. URL: <https://doi.org/10.32628/cseit2173105> (дата звернення: 12.11.2024).
2. Sarker I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*. 2021. Т. 2, № 3. URL: <https://doi.org/10.1007/s42979-021-00592-x> (дата звернення: 11.11.2024).
3. Bergmann D. What Is Semi-Supervised Learning? | IBM. *IBM - United States*. URL: <https://www.ibm.com/topics/semi-supervised-learning#:~:text=Semi-supervised%20learning%20is%20a,for%20classification%20and%20regression%20tasks>. (дата звернення: 11.11.2024).

Візуалізація багатомірної векторної бази даних

*Товстенко Максим Андрійович,
студент групи ПД-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
tovsmax@gmail.com*

*Науковий керівник: Садовенко Володимир Сергійович,
кандидат фізико-математичних наук, доцент,
професор кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Поширення великих мовних моделей (ВММ) створило нові проблеми, зокрема, щодо релевантності та надійності згенерованого тексту на основі їхніх навчальних даних. Векторні бази даних пропонують рішення, розширюючи контекст ВММ за допомогою попередньо відібраної, релевантної та надійної інформації. Однак виникає

критичне питання: «Які дані з векторної бази даних слід інтегрувати в контекст?» Незважаючи на існування численних алгоритмів пошуку релевантної інформації, ефективні методи візуалізації є важливими для розуміння та оптимізації цього процесу інтеграції.

Постановка задачі.

Задачею дослідження є розробка методик для візуального відлагодження інтерактивних інформаційно-допоміжних систем на основі великих мовних моделей та векторних баз даних.

Мета дослідження.

Метою цього дослідження є оцінка та вибір найбільш ефективних алгоритмів зменшення розмірності для візуалізації простору векторного вбудовування у векторній базі даних, які зберігають довідкову інформацію. Розглянуті алгоритми включають t-розподілене вбудовування стохастичних сусідів (t-SNE), рівномірну множинну апроксимацію та проєкцію (UMAP) та аналіз головних компонент (PCA).

Результати дослідження.

Попередні результати дослідження показали, що алгоритми зменшення розмірності, такі як t-SNE, UMAP та PCA, дозволяють візуалізувати багатовимірний простір векторних вбудовувань для покращення інтеграції даних у контекст ВММ. UMAP і t-SNE демонструють високу ефективність у збереженні локальної структури та семантичних зв'язків між векторами, тоді як PCA менш придатний для складних структур. Візуалізації допомагають виявляти релевантні кластери та ізольовані точки, що сприяє оптимізації вибору інформації. Розроблені інтерактивні прототипи спрощують аналіз векторів, налаштування параметрів пошуку та покращення точності генерацій тексту ВММ, що відкриває перспективи для автоматизації процесу інтеграції.

Висновки та перспективи.

Візуалізація багатовимірного простору вбудовування забезпечує глибше розуміння контекстних рішень ВММ. Сегментуючи контекст і згенерований текст на фрагменти, перетворюючи їх на вектори вбудовування і візуалізуючи їх у зменшеному просторі вбудовування, ми можемо краще зрозуміти причини, що лежать в основі конкретних результатів генерації тексту. Такий підхід розширює наші можливості вдосконалення та оптимізації використання векторних баз даних у поєднанні з великими мовними моделями.

Список використаних джерел:

1. Merritt R. What is retrieval-augmented generation aka rag?. NVIDIA Blog. URL: <https://web.archive.org/web/20240510044450/https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/> (date of access: 08.06.2024).
2. McInnes L., Healy J., Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2020. 63 p. (Preprint. Tutte Institute for Mathematics and Computing ; 1802.03426). URL: <https://doi.org/10.48550/arXiv.1802.03426> (date of access: 08.06.2024).
3. Salton G., Wong A., Yang C. S. A vector space model for automatic indexing. *Communications of the ACM*. 1975. Vol. 18, no. 11. P. 613–620. URL: <https://doi.org/10.1145/361219.361220> (date of access: 24.11.2024).
4. Pan J. J., Wang J., Li G. Survey of vector database management systems. *The VLDB Journal*. 2024. URL: <https://doi.org/10.1007/s00778-024-00864-x> (date of access: 24.11.2024).
5. Salton G., Wong A., Yang C. S. A vector space model for automatic indexing. *Communications of the ACM*. 1975. Vol. 18, no. 11. P. 613–620. URL: <https://doi.org/10.1145/361219.361220> (date of access: 24.11.2024).
6. Taipalus T. Vector database management systems: fundamental concepts, use-cases, and current challenges. *Cognitive systems research*. 2024. Vol. 85. P. 101216. URL: <https://doi.org/10.1016/j.cogsys.2024.101216> (date of access: 24.11.2024).

Штучний інтелект як фінансовий порадник: подолання бар'єрів довіри та адаптація до потреб користувача

*Федоренко Максим Леонідович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
fedorenko.mx@gmail.com
Науковий керівник: Замрій Ірина Вікторівна,
доктор технічних наук, доцент,
завідувач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Незважаючи на те, що штучний інтелект демонструє високий потенціал у фінансовій сфері, рівень довіри до нього серед користувачів залишається низьким. Багато людей хвилюються через можливість помилок у прогнозах, втрату конфіденційності даних чи складність у взаємодії з технологією [2]. У цьому контексті важливо не лише вдосконалювати технології, а й адаптувати їх до реальних потреб користувачів, щоб подолати бар'єри недовіри.

Постановка задачі.

Задачею дослідження є розробка системи фінансових рекомендацій на основі ШІ, яка враховує основні проблеми довіри користувачів та пропонує зрозумілу й безпечну взаємодію з системою.

Мета дослідження.

Метою дослідження є визначення шляхів підвищення довіри до ШІ у фінансовій сфері шляхом впровадження прозорих механізмів роботи, адаптації під потреби користувача та забезпечення захисту конфіденційності даних.

Результати дослідження.

Дослідження показало, що для підвищення довіри до штучного інтелекту у фінансовій сфері необхідно впроваджувати комплексний підхід, який включає прозорість, безпеку та адаптивність [3]. Прототип розробленої системи демонструє, як ШІ може пояснювати свої рішення користувачеві через спеціально створений

інтерфейс аналітики. Кожна рекомендація супроводжується простим і зрозумілим поясненням, що дає змогу користувачеві розуміти логіку роботи алгоритму. Значну увагу приділено захисту даних: застосовані сучасні протоколи шифрування гарантують, що особиста інформація користувачів залишається конфіденційною. [4] Крім того, система дозволяє адаптувати рівень автоматизації під уподобання користувача. Наприклад, користувач може самостійно приймати остаточні рішення, базуючись на рекомендаціях ШІ, або ж повністю довіряти автоматизованому процесу [5].

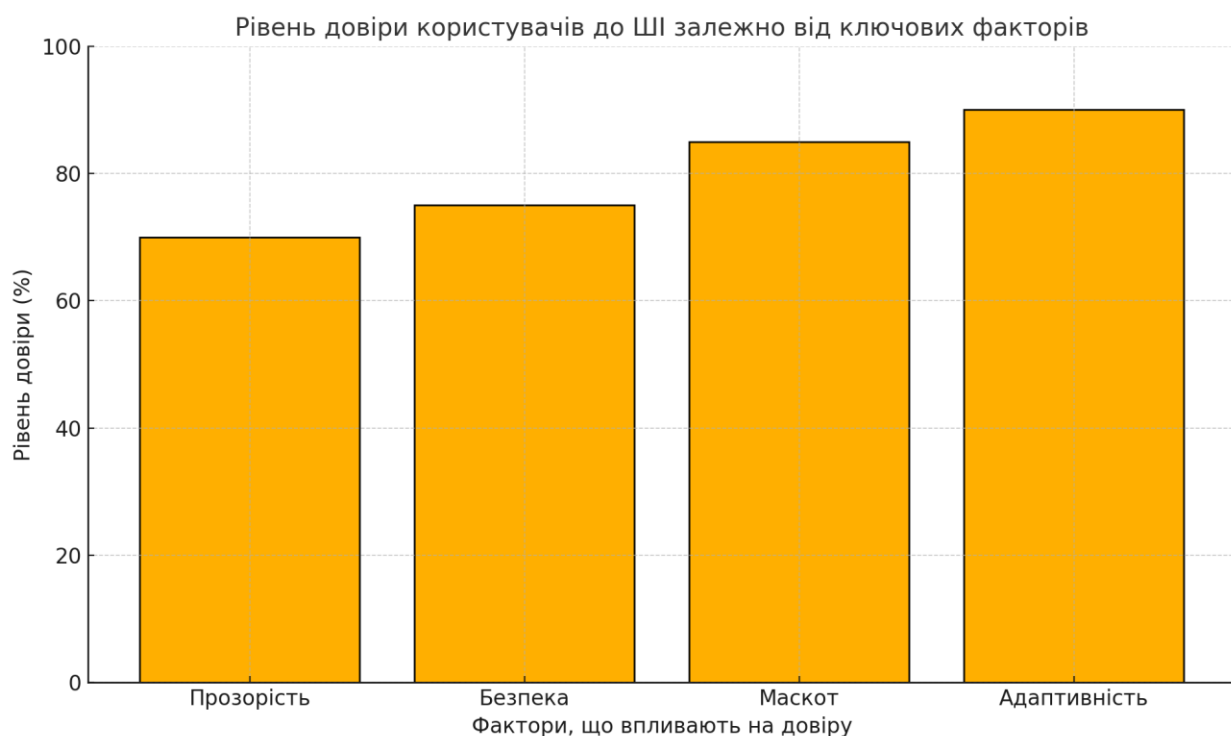


Рис. 1. Рівень довіри користувачів до ШІ

Для створення більш дружньої атмосфери взаємодії впроваджено маскота – віртуального персонажа, який супроводжує користувача під час використання додатку. Маскот не лише допомагає у спілкуванні з системою, але й створює емоційно приємний досвід, піднімаючи настрій та зменшуючи рівень напруги під час вирішення складних фінансових питань. Він також виконує роль своєрідного «посередника», допомагаючи користувачам краще зрозуміти можливості ШІ. Результати дослідження свідчать, що така система не лише спрощує взаємодію з технологією, але й змінює

сприйняття штучного інтелекту, формуючи довіру та впевненість у його рекомендаціях. Завдяки інтуїтивно зрозумілому дизайну та елементам емоційної підтримки, користувачі відчувають, що ШІ може стати їхнім справжнім партнером у фінансових питаннях [6].

Висновки та перспективи.

Підвищення довіри до ШІ у фінансовій сфері є важливим фактором для його широкого впровадження. Основні висновки дослідження:

Користувачі більше довіряють ШІ, коли мають доступ до пояснення рішень.

Персоналізований підхід і гнучкість у взаємодії сприяють зменшенню бар'єрів довіри.

Забезпечення конфіденційності даних залишається ключовим пріоритетом.

Перспективи включають впровадження технологій блокчейну для підвищення прозорості транзакцій, створення адаптивних моделей для роботи з різними категоріями користувачів та вдосконалення емоційного інтелекту ШІ для покращення взаємодії з людиною.

Список використаних джерел:

1. What is artificial intelligence (AI) in finance? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/topics/artificial-intelligence-finance>.
 2. Can We Trust Artificial Intelligence? [Електронний ресурс] – Режим доступу до ресурсу: <https://scienceexchange.caltech.edu/topics/artificial-intelligence-research/trustworthy-ai>.
 3. How to design an AI interface users will trust in 2024? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lazarev.agency/articles/how-to-design-an-ai-interface-users-will-trust-in-2024>.
 4. Top 5 Ways to Protect and Secure Data in the Age of AI [Електронний ресурс] – Режим доступу до ресурсу: <https://www.publicissapient.com/insights/data-security-for-ai>.
 5. AI Will Understand Humans Better Than Humans Do [Електронний ресурс] – Режим доступу до ресурсу: <https://www.wired.com/story/plaintext-ai-will-understand-humans-better-than-humans-do/>.
- How AI-powered mascots are changing the brand communication game [Електронний ресурс] – Режим доступу до ресурсу: <https://campaignme.com/how-ai-powered-mascots-are-changing-the-brand-communication-game/>.

Автоматизоване машинне навчання: прийняття рішень на основі штучного інтелекту у бізнес аналітиці

*Юр'єв Андрій Ленонідович,
студент групи ІСДМ-64,
спеціальність 126 Інформаційні системи та технології,
Державного університету інформаційно-комунікаційних технологій
andreyuyryev@ukr.net*

*Шахматов Іван Олександрович
викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій
ivan.shakhmatov@gmail.com*

*Науковий керівник: Сватко Віталій Володимирович,
кандидат технічних наук, доцент,
Державного університету інформаційно-комунікаційних технологій*

Автоматизоване машинне навчання (AutoML) є потужним інструментом, який дозволяє бізнесам використовувати штучний інтелект (ШІ) для прийняття рішень. Це особливо важливо в умовах швидко змінюваного бізнес-середовища, де дані стають основним активом. AutoML спрощує процес розробки моделей машинного навчання, роблячи його доступним навіть для фахівців, які не мають глибоких знань у цій галузі.

Постановка задачі.

У рамках даного дослідження ставиться задача вивчення впливу автоматизованого машинного навчання (AutoML) на процеси прийняття рішень у бізнес-аналітиці. Зокрема, дослідження спрямоване на аналіз переваг та недоліків використання AutoML у бізнес-середовищі, а також оцінку його впливу на швидкість і точність прийняття рішень. Важливим є визначення галузей, які найбільше виграють від впровадження AutoML, та виявлення викликів, з якими стикаються бізнеси при інтеграції цієї технології у свої процеси. Результати дослідження мають на меті надати ґрунтовне розуміння ролі AutoML у сучасній бізнес-аналітиці та сприяти ефективному використанню штучного інтелекту для прийняття стратегічних рішень.

Мета дослідження.

Метою даного дослідження є аналіз впливу автоматизованого машинного навчання (AutoML) на ефективність бізнес-аналітики. Дослідження також спрямоване на визначення ключових факторів, що сприяють успішному впровадженню AutoML у бізнес-процеси, а також на оцінку перспектив розвитку AutoML у контексті сучасних бізнес-тенденцій.

Результати дослідження.

Дослідження впливу автоматизованого машинного навчання (AutoML) на бізнес-аналітику виявило ряд важливих аспектів, які варто розглянути детальніше.

Однією з головних переваг AutoML є зниження витрат на розробку моделей машинного навчання. Традиційно, створення таких моделей вимагало залучення висококваліфікованих спеціалістів з даних, що супроводжувалося значними фінансовими витратами. AutoML автоматизує процеси підбору алгоритмів, налаштування гіперпараметрів та валідації моделей, що дозволяє суттєво скоротити час та ресурси, необхідні для розробки.

Другою вагомою перевагою є прискорення процесу аналізу даних і прийняття рішень. Завдяки автоматизації рутинних завдань, аналітики можуть швидше отримувати інсайти з даних та оперативно реагувати на змінні умови ринку. Це особливо важливо в сучасному бізнес-середовищі, де швидкість прийняття рішень може стати вирішальним фактором успіху.

Третя перевага полягає у підвищенні точності прогнозів завдяки використанню складних алгоритмів. AutoML системи здатні автоматично обирати найбільш підходящі моделі та оптимізувати їх параметри, що може призвести до покращення якості прогнозів. Це відкриває можливості для більш точного планування та стратегічного управління.

Незважаючи на значні переваги, використання AutoML супроводжується певними викликами. Перш за все, існує залежність від якості вхідних даних. AutoML не може компенсувати погану якість або нерелевантних даних; якщо дані містять

помилки або не відображають сутності проблеми, результати моделювання будуть неточними. Тому важливо забезпечити якісний збір та підготовку даних перед використанням AutoML.

Другий виклик стосується проблем з прозорістю моделей, що ускладнює їх інтерпретацію. Багато алгоритмів, які використовуються в AutoML, є "чорними ящиками", що робить важким пояснення, як саме модель прийшла до певного висновку. Це може бути критичним у галузях, де необхідно пояснювати рішення, наприклад, у фінансовому або медичному секторах.

Третій недолік пов'язаний з необхідністю наявності експертів для аналізу результатів. Хоча AutoML автоматизує багато аспектів моделювання, інтерпретація та застосування результатів все ще вимагають експертних знань. Без належного розуміння, як використовувати отримані інсайти, бізнес може не повністю реалізувати потенціал технології.

AutoML знайшов широке застосування у різних секторах економіки. У фінансовому секторі технологія використовується для прогнозування ризиків та виявлення шахрайства. Автоматизація аналізу великих обсягів транзакційних даних дозволяє банкам та фінансовим установам швидше виявляти підозрілі активності та приймати відповідні заходи.

У сфері роздрібної та оптової торгівлі AutoML сприяє персоналізації пропозицій та оптимізації запасів. Аналіз поведінки покупців та прогнозування попиту допомагають компаніям ефективніше керувати ланцюгами постачання та підвищувати рівень задоволеності клієнтів.

В галузі охорони здоров'я AutoML використовується для прогнозування захворювань та аналізу медичних даних. Це дозволяє медичним закладам покращувати діагностику, планувати ресурси та надавати більш індивідуалізоване лікування пацієнтам.

Перспективи розвитку AutoML є значними. По-перше, очікується зростання популярності AutoML у малих і середніх підприємствах. Завдяки спрощенню процесів

машинного навчання, навіть компанії з обмеженими ресурсами можуть впроваджувати передові аналітичні рішення.

По-друге, відбувається інтеграція AutoML з іншими технологіями, такими як Інтернет речей (IoT) та великі дані. Це відкриває нові можливості для збору та аналізу даних у реальному часі, що може призвести до створення більш адаптивних та інтелектуальних систем управління.

По-третє, важливим напрямком є розвиток нових методів для підвищення прозорості моделей. Робота в цьому напрямку сприятиме підвищенню довіри до AutoML-рішень та їх ширшому впровадженню, особливо в регульованих галузях, де пояснюваність моделей є критичною.

Висновки та перспективи

Автоматизоване машинне навчання має значний потенціал для трансформації бізнес-аналітики, забезпечуючи швидкість, точність і доступність аналітичних рішень. Однак, для успішного впровадження AutoML необхідно враховувати виклики, пов'язані з якістю даних і прозорістю моделей. Перспективи розвитку AutoML виглядають обнадійливо, з можливістю інтеграції в різні галузі та адаптації до нових технологічних умов.

Список використаних джерел:

1. Hutter, F., Kotthoff, L., Vanschoren, J. Automated Machine Learning: Methods, Systems, Challenges. – Springer, 2019. – [Електронний ресурс]. – Режим доступу: <https://link.springer.com/book/10.1007/978-3-030-05318-5>.
2. Feurer, M., Hutter, F. Hyperparameter Optimization // Automated Machine Learning. – Springer, 2019. – С. 3–33. – [Електронний ресурс]. – Режим доступу: https://link.springer.com/chapter/10.1007/978-3-030-05318-5_1.
3. Xin He, Kaiyong Zhao, Xiaowen Chu. AutoML: A survey of the state-of-the-art // Knowledge-Based Systems. – 2021. – DOI: 10.1016/j.knosys.2020.106622.
4. Jayesh Rane, Suraj Kumar Mallick, Ömer Kaya, Nitin Liladhar Rane. Automated Machine Learning (AutoML) in industry 4.0, 5.0, and society 5.0: Applications, opportunities, challenges, and future directions. – 2024. – DOI: 10.70593/978-81-981271-0-5_5.

Метод аналізу продуктивності ORM-технологій у веб-застосунках на С# з використанням ASP.NET: Порівняльне дослідження на основі типових операцій з базою даних

*Побережник Андрій Анатолійович
студент 6 курсу, групи ПДМ-62*

*Державного університету інформаційно-комунікаційних технологій
andrewdev2021@gmail.com*

*Науковий керівник: Соляник Людмила Олесіївна,
к.х.н., доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Постановка задачі

Використання технологій ORM (Object-Relationship Mapping) реалізовано з базами даних, написаними на С# з використанням ASP.NET. Технології ORM забезпечують високий ступінь абстракції, дозволяючи розробникам взаємодіяти з базою даних через об'єкти програмування, та через запити SQL. Таким чином, різні методи ORM можна адаптувати відповідно до типу програми, типу даних і формату даних. Правильний вибір ORM є важливим для досягнення високої продуктивності, масштабованості та часу відгуку веб-додатків [1], [2]. Тому необхідно проаналізувати складність на основі вимог ORM, щоб зробити правильний вибір.

Мета дослідження

Метою даного дослідження є проведення порівняльного аналізу продуктивності популярних ORM-рішень у веб-застосунках на С# з використанням ASP.NET. Основну увагу приділено аналізу Entity Framework, Dapper та інших ORM-технологій, що широко застосовуються у розробці. Дослідження охоплює оцінку їх продуктивності при виконанні стандартних CRUD-операцій (створення, читання, оновлення та видалення) з базою даних. Особливу увагу зосереджено на таких аспектах, як швидкість виконання операцій, споживання пам'яті та зручність реалізації складних запитів .

Результати дослідження

Дослідження показало суттєві відмінності у продуктивності різних ORM-рішень залежно від типу виконуваних операцій та розміру оброблюваних даних. Entity Framework продемонстрував високу зручність у використанні, завдяки чому

знижуються витрати часу на розробку, але у деяких сценаріях виявився менш продуктивним порівняно з Dapper. Dapper, це мікро-ORM, який забезпечує вищу швидкість виконання операцій завдяки прямому доступу до SQL-запитів. Проте він вимагає більшого обсягу коду та не дає всіх можливостей, властивих ORM. Результати тестування також демонструють залежність продуктивності від складності операцій, зокрема, Dapper переважає при роботі з великими обсягами даних, тоді як Entity Framework забезпечує кращу інтеграцію з іншими інструментами ASP.NET.

Висновки та перспективи

Даний підхід до аналізу продуктивності ORM-технологій дозволяє розробникам веб-застосунків на C# обрати оптимальне рішення залежно від специфіки проекту. Entity Framework є гарним вибором для невеликих або середніх проєктів із типовими запитамі, тоді як Dapper підходить для високонавантажених систем або задач, що потребують максимальної продуктивності [3], [4]. Подальші дослідження будуть зосереджені на оцінці ефективності ORM-технологій в умовах високих навантажень, роботи з масштабованими базами даних та інтеграції з хмарними сервісами [5].

Список використаних джерел:

- 1.C# Guide - .NET managed language. *Microsoft Learn: Build skills that open doors in your career.* URL: <https://learn.microsoft.com/en-my/dotnet/csharp/>
- 2.ASP.NET Core | Open-source web framework for .NET. *Microsoft.* URL: <https://dotnet.microsoft.com/en-us/apps/aspnet>
- 3.ORM. *Flexberry PLATFORM Documentation.* URL: https://flexberry.github.io/ru/gbt_orm.html
- 4.Abba I. What is an ORM – The Meaning of Object Relational Mapping Database Tools. *freeCodeCamp.org.* URL: [https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/#:~:text=Object%20Relational%20Mapping%20\(ORM\)%20is,\(OOD\)%20to%20relational%20databases.](https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/#:~:text=Object%20Relational%20Mapping%20(ORM)%20is,(OOD)%20to%20relational%20databases.)
- 5.Awati R. What is object-relational mapping (ORM)? – TechTarget Definition. *TheServerSide.com.* URL: <https://www.theserverside.com/definition/object-relational-mapping-ORM>

Аналіз сучасних засобів програмної інженерії для розробки експертних систем

*Лукашенко Ігор Петрович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
egorkaluki46@gmail.com
Науковий керівник: Корецька Вікторія Олександрівна,
к.пед.н., доцент, професор кафедри ІТ
Державного університету інформаційно-комунікаційних технологій*

Медична галузь все частіше використовує експертні системи на основі штучного інтелекту (ШІ) для покращення процесів прийняття рішень.[1, 3] Ефективна реалізація цих систем значною мірою залежить від передових технологій програмної інженерії, що охоплюють платформи машинного навчання (ML), інструменти управління даними, хмарні сервіси та системи баз даних. Це дослідження оцінює ключові інструменти програмної інженерії, які дозволяють розробляти експертні системи.

Постановка задачі.

Дослідження спрямоване на визначення ідеальної комбінації засобів програмної інженерії для побудови експертної системи, що забезпечує точність, ефективність та надійність при обробці медичної інформації.

Мета дослідження.

Мета полягає в оцінці ефективності сучасних програмних рішень для створення систем підтримки прийняття рішень, з особливим акцентом на первинну медичну допомогу, де обробка різноманітних типів даних має важливе значення.

Результати дослідження.

1. Фреймворки та бібліотеки штучного інтелекту:
 - а. TensorFlow та PyTorch пропонують модульні та надійні інструменти для побудови глибоких нейронних мереж [2,7]. Scikit-learn

полегшує традиційні методи ML, які чудово підходять для класифікації медичних даних та маніпулювання даними.

2. Для роботи з даними:

a. Pandas і NumPy широко використовуються для очищення та аналізу даних[4, 5].

b. Hugging Face Transformers ідеальні для роботи з текстовими даними, наприклад, медичними записами[6] .

3. Системи зберігання даних: бази даних SQL та NoSQL забезпечують ефективне зберігання як структурованої, так і неструктурованої інформації.

4. Метрики оцінювання: такі показники, як точність, показник F1 та AUC-ROC, мають вирішальне значення для перевірки продуктивності моделі, що було продемонстровано в експерименті, який показав 96% точності.

Висновки та перспективи.

Проведений аналіз показує, що використання сучасних фреймворків (TensorFlow, PyTorch), інструментів обробки даних (Pandas, Hugging Face), баз даних (SQL, NoSQL) та хмарних сервісів є критично важливим для успішного впровадження експертних систем. Подальші дослідження можуть бути зосереджені на оптимізації ресурсів та розробці систем реального часу, що покращить їх впровадження на практиці.

Список використаних джерел:

1. Deep Learning in Medicine. Are We Ready? - PubMed / D. SW Ting et al. PubMed. URL: <https://pubmed.ncbi.nlm.nih.gov/30788488/> (date of access: 20.11.2024).

Guide | TensorFlow Core. TensorFlow. URL: <https://www.tensorflow.org/guide> (date of access: 20.11.2024).

2. J. Topol E. High-performance medicine: the convergence of human and artificial intelligence - Nature Medicine. Nature. URL: <https://www.nature.com/articles/s41591-018-0300-7> (date of access: 20.11.2024).

3. NumPy documentation – numpy v2.1 manual. NumPy -. URL: <https://numpy.org/doc/stable/> (date of access: 20.11.2024).
4. Pandas documentation – pandas 2.2.3 documentation. pandas - Python Data Analysis Library. URL: <https://pandas.pydata.org/docs/> (date of access: 20.11.2024).
5. Transformers. Hugging Face – The AI community building the future. URL: <https://huggingface.co/docs/transformers/index> (date of access: 20.11.2024).
6. Welcome to PyTorch Tutorials – PyTorch Tutorials 2.5.0+cu124 documentation. PyTorch. URL: <https://pytorch.org/tutorials/> (date of access: 20.11.2024).

Алгоритм впровадження шучного інтелекту у процес написання коду на прикладі chatgpt

*Петрунчак Анна Романівна,
студентка групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
anpnboyko@gmail.com*

*Науковий керівник: Довженко Тимур Павлович,
кандидат технічних наук,
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

У сучасному програмуванні штучний інтелект (ШІ) починає активно застосовуватись для полегшення процесу написання коду. ChatGPT, як потужний інструмент на базі GPT-4, здатний автоматизувати багато аспектів програмування, надаючи можливість генерації коду, його вдосконалення та тестування. У роботі буде розглянуто алгоритм інтеграції ChatGPT у процес написання коду, а також оцінка ефективності цього підходу для розробників.

Мета дослідження:

Дослідити та розробити алгоритм впровадження ChatGPT у процес написання програмного коду для підвищення продуктивності розробників та автоматизації рутинних задач програмування.

Предмет дослідження:

Процес інтеграції ChatGPT у розробку програмного коду та його вплив на ефективність програмування.

Постановка задачі:

Задачею дослідження є створення алгоритму впровадження ChatGPT у процес програмування, аналіз його здатності до генерації, вдосконалення та тестування коду, а також оцінка впливу на ефективність розробки програмного забезпечення.

Результати дослідження:

В результаті дослідження розроблено алгоритм взаємодії з ChatGPT для автоматизації написання програмного коду. Визначено, що використання ChatGPT значно зменшує час на виконання рутинних завдань, таких як написання шаблонного коду, а також дозволяє швидше знаходити та виправляти помилки у коді. Інтеграція цієї технології також дозволяє розробникам зосередитись на більш складних аспектах програмування, покращуючи продуктивність і якість програмного забезпечення.

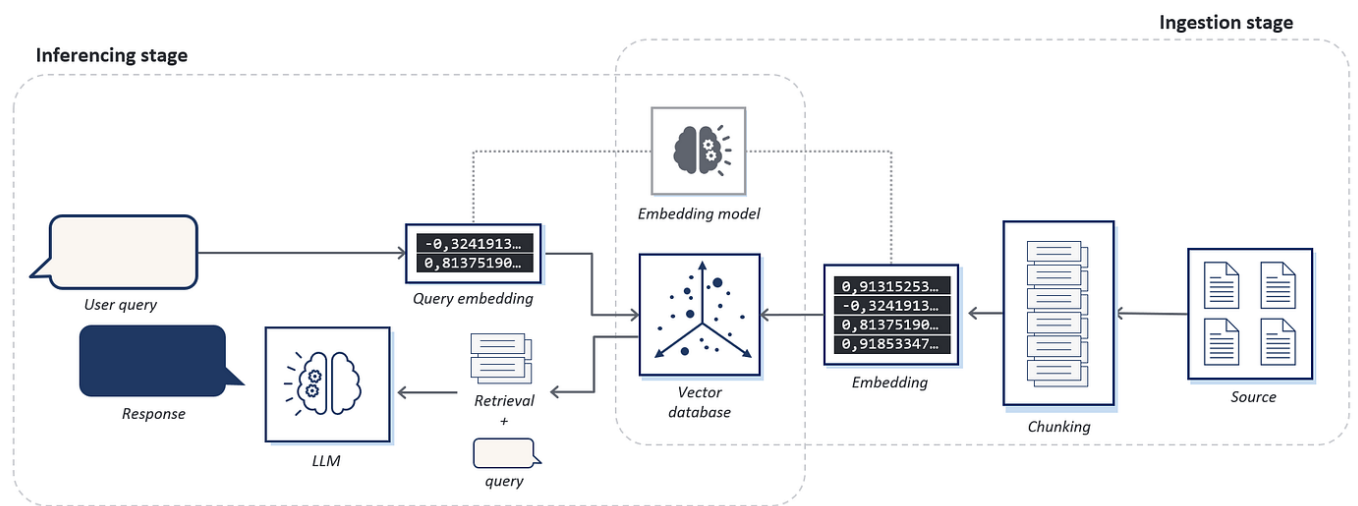


Рис. 1. Схема підключення до ChatGPT через API

Висновки та перспективи:

Впровадження ChatGPT у процес написання коду дозволяє значно покращити ефективність розробки програмного забезпечення, зменшивши витрати часу на виконання рутинних завдань. Подальші дослідження можуть бути спрямовані на удосконалення інтеграції ChatGPT з іншими інструментами програмування, а також на вивчення його застосування для автоматизованого тестування коду.

Список використаних джерел:

1. “The Complete Guide: How to Use ChatGPT API in Application” 2024. – [Електронний ресурс] – Режим доступу: <https://adamfard.com/blog/how-to-use-chatgpt-api>
2. “GPT-3 Coding Challenges: Overcoming Barriers with AI” 2024. – [Електронний ресурс] – Режим доступу: <https://appmaster.io/blog/gpt-3-coding-challenges>
3. “How to Use GPT-4 for Writing Code: A Beginner's Guide” 2024. – [Електронний ресурс] – Режим доступу: <https://medium.com/@jonoyanguren/my-adventure-with-openai-a-beginners-guide-to-coding-with-the-gpt-4-api-in-python-a7de87e7c569>
4. "The Future of AI in Software Development”, 2024. . – [Електронний ресурс] – Режим доступу: <https://fortyseven47.com/blog/the-future-of-ai-in-software-development/>

Визначення пріоритетності пацієнтів у відділенні невідкладної допомоги як задача машинного навчання

*Спіцин Андрій Ярославович,
студент 6 курсу, групи ПДМ-61
Державного університету інформаційно-комунікаційних технологій
and.spitsyn@gmail.com*

*Науковий керівник: Садовенко Володимир Сергійович,
кандидат фізико-математичних наук, доцент,
професор кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Відділення невідкладної допомоги або ВНД є важливим елементом системи охорони здоров'я, оскільки саме тут пацієнти отримують невідкладну медичну допомогу у разі екстрених станів, травм або інших небезпечних для життя ситуацій. Зважаючи на тренд зростання кількості пацієнтів і високу динаміку медичних випадків [1], проблема визначення пріоритетності пацієнтів у ВНД є актуальною, оскільки правильний розподіл ресурсів та своєчасність медичних рішень можуть значно впливати на результати лікування та загальний стан здоров'я пацієнтів.

З кожним роком кількість звернень до ВНД очікувано зростає через збільшення населення, старіння суспільства, підвищення частоти хронічних захворювань та вплив зовнішніх чинників, таких як аварії, катастрофи чи навіть пандемії. Це призводить до перевантаження медичного персоналу та обмеженості ресурсів, що ускладнює надання своєчасної допомоги всім пацієнтам. У таких умовах визначення пріоритетності стає одним із найважливіших механізмів оптимізації потоку пацієнтів і розподілу наявних ресурсів, аби найважчі випадки отримали невідкладну допомогу, тоді як менш критичні можуть зачекати.

Неправильна оцінка пріоритетності може мати серйозні наслідки для здоров'я та життя пацієнтів [2]. У випадках, коли пацієнти з критичними станами не отримують своєчасної допомоги ризикують зіткнутися з ускладненнями або навіть летальними наслідками. Натомість, якщо пацієнти з менш серйозними проблемами отримують допомогу раніше за критично хворих, це призводить до неефективного використання ресурсів і затримки для тих, хто дійсно потребує невідкладного втручання. Таким чином, якість і точність процесу визначення пріоритетності безпосередньо впливає на результати лікування та загальний рівень безпеки пацієнтів.

Крім того, сам процес визначення пріоритетності пацієнтів займає час. Традиційний процес тріажу, який ґрунтується на рішеннях медичного персоналу, часто потребує значного часу на обробку кожного окремого випадку, особливо коли відділення перевантажене пацієнтами з різними рівнями тяжкості стану. Через це має сенс розробити метод для автоматизації цього процесу. Таким методом може стати алгоритм машинного навчання, що приймає життєві показники пацієнта разом із первинним діагнозом та може миттєво визначати його пріоритетність [3]. Разом із цим, дозволивши виконувати цей процес автоматизованій системі, це повністю позбавить процес людського фактору, що при правильних налаштуваннях буде давати оптимальні оцінки пацієнтів кожного разу.

Іншим фактором є обмежені ресурси, такі як медичний персонал, обладнання та лікарняні ліжка, які є постійною проблемою для більшості закладів охорони здоров'я.

Оптимальні оцінки пріоритетності пацієнтів дозволять ефективніше використовувати ці ресурси, надаючи невідкладну допомогу тим, хто потребує її негайно. Застосування моделей на основі машинного навчання дозволять швидше та точніше визначити потреби кожного пацієнта, що знижує навантаження на персонал і скорочує час очікування для критично хворих пацієнтів.

Висновок

Завдання визначення пріоритетності пацієнтів у відділенні невідкладної допомоги є актуальним, бо його вирішення здатне значно підвищити ефективність надання медичної допомоги та забезпечити справедливий і оптимальний розподіл ресурсів у лікарнях. Автоматизація та використання методів машинного навчання у тριαжі є перспективним напрямком, який дозволить поліпшити точність і швидкість прийняття рішень, мінімізувати ризики, пов'язані з людським фактором, і знизити загальне навантаження на систему охорони здоров'я.

Список використаних джерел:

1. Roberts D. C., McKay M. P., Shaffer A. Increasing Rates of Emergency Department Visits for Elderly Patients in the United States, 1993 to 2003. *Annals of Emergency Medicine*. 2008. Т. 51, № 6. С. 769–774. URL: <https://doi.org/10.1016/j.annemergmed.2007.09.011> (дата звернення: 13.11.2024).
2. Suamchaiyaphum K., Jones A. R., Markaki A. Triage Accuracy of Emergency Nurses: An Evidence-Based Review. *Journal of Emergency Nursing*. 2023. URL: <https://doi.org/10.1016/j.jen.2023.10.001> (дата звернення: 13.11.2024).
3. Machine learning-based models to support decision-making in emergency department triage for patients with suspected cardiovascular disease / H. Jiang та ін. *International Journal of Medical Informatics*. 2021. Т. 145. С. 104326. URL: <https://doi.org/10.1016/j.ijmedinf.2020.104326> (дата звернення: 11.11.2024).

Аналіз ефективності використання нейронних мереж для розпізнавання емоцій в голосових повідомленнях

*Левченко Олександр Олександрович,
студент групи ПДМ-62
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
tag6269128@stud.duikt.edu.ua
Шахматов Іван Олександрович
Викладач кафедри ІІЗ
Державного університету інформаційно-комунікаційних технологій
ivan.shakhmatov@gmail.com
Науковий керівник: Довженко Тимур Павлович,
доцент кафедри, кандидат технічних наук,
Державного університету інформаційно-комунікаційних технологій*

Постановка задачі.

Розпізнавання емоцій у голосових повідомленнях є актуальною проблемою для різних сфер, включаючи психологічну підтримку, автоматизовані гарячі лінії та персоналізовані сервіси. Використання нейронних мереж, як перспективного інструменту, дозволяє покращити точність визначення емоцій завдяки їх здатності обробляти складні патерни в акустичних характеристиках. Основною задачею є аналіз ефективності цих підходів на основі наявних наукових праць і результатів досліджень.

Мета дослідження.

Метою дослідження є детальний аналіз ефективності використання нейронних мереж для розпізнавання емоцій у голосових повідомленнях. Це включає систематизацію сучасних підходів до застосування нейронних мереж у цій галузі, вивчення їх можливостей у порівнянні з традиційними методами та оцінку їх продуктивності в різних сценаріях.

Дослідження спрямоване на визначення переваг та недоліків нейронних мереж при обробці складних акустичних патернів, що відображають емоційний стан мовця. Крім того, метою є розробка рекомендацій щодо оптимізації моделей нейронних мереж для підвищення точності та надійності розпізнавання емоцій у практичних застосуваннях, таких як психологічна підтримка, автоматизовані гарячі лінії та персоналізовані сервіси.

Результати дослідження.

У результаті аналізу літератури та наукових джерел встановлено, що сучасні нейронні мережі є одним із найефективніших інструментів для розпізнавання емоцій у голосових повідомленнях. Рекурентні нейронні мережі (RNN) з компонентами довготривалої короткочасної пам'яті (LSTM) активно використовуються для аналізу часових залежностей у мовленні та показують точність розпізнавання на рівні 80–83% на датасетах, таких як RAVDESS. Конволюційні нейронні мережі (CNN) демонструють високу ефективність у виділенні спектральних ознак голосу, досягаючи точності до 82% (Рис.1).

Гібридні моделі, які поєднують CNN та LSTM, об'єднують переваги обох підходів і досягають найкращих результатів, з точністю 85–87% при аналізі емоцій, таких як радість, сум, гнів і нейтральний стан. Порівняння з традиційними методами, такими як машини опорних векторів (Support Vector Machines, SVM), показало, що останні поступаються нейронним мережам, демонструючи точність у межах 70–75%. Отже, нейронні мережі, особливо гібридні моделі, є перспективним напрямом для підвищення точності розпізнавання емоцій у голосових повідомленнях.

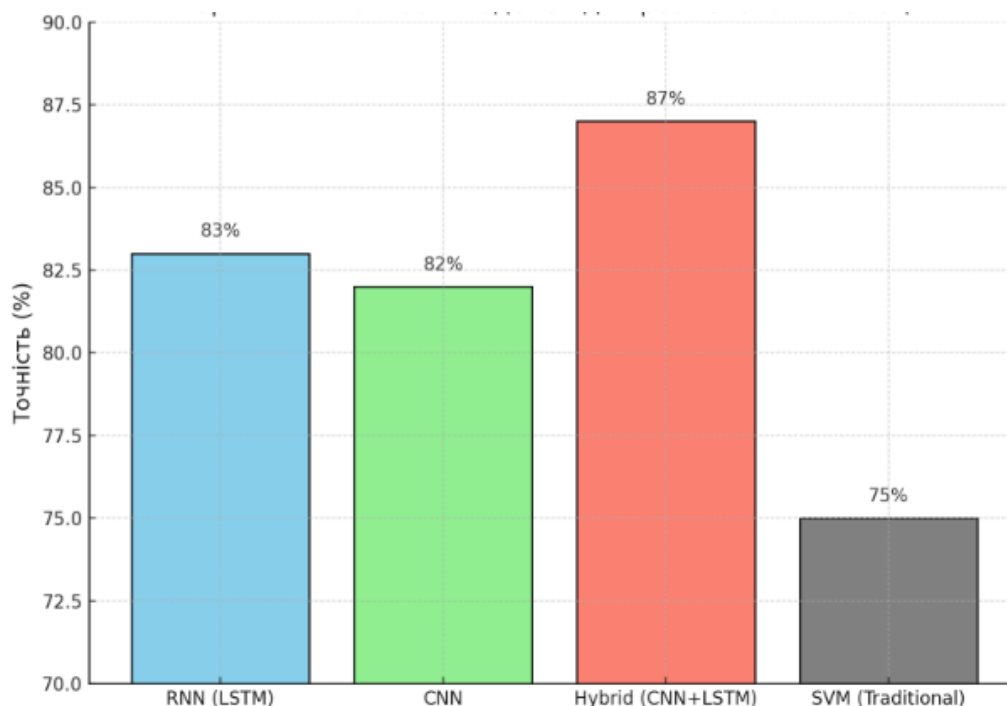


Рис.1. Графік порівняння

Висновки та перспективи.

Нейронні мережі, завдяки їхній здатності обробляти складні нелінійні залежності, показують високу ефективність у розпізнаванні емоцій у голосових повідомленнях. Їх застосування дозволяє підвищити точність у порівнянні з традиційними методами. У перспективі важливо дослідити можливості інтеграції нейронних мереж із мультимодальними системами, які комбінують голосові, текстові та візуальні дані, а також адаптувати їх для роботи в реальному часі.

Список використаних джерел:

1. Larionov Oleg, Anton Astakhov, Anna Krasovska, and Iryna Domanetska. "Intelligent module for recognizing emotions by voice." *Advanced Information Technology*, no. 1 (1) (2021): 46–52. [Електронний ресурс] – Режим доступу: <http://dx.doi.org/10.17721/ait.2021.1.06>.
2. Farhad Morteza pour Shiri, Thinagaran Perumal, Norwati Mustapha, Raihani Mohamed "A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU". *arXiv.org*. [Електронний ресурс] – Режим доступу: <https://arxiv.org/abs/2305.17473>.
3. Eva Lieskovská, Maroš Jakubec, Roman Jarina ORCID, Michal Chmulík "Review on Speech Emotion Recognition Using Deep Learning and Attention Mechanism". *MDPI*. [Електронний ресурс] – Режим доступу: <https://www.mdpi.com/2079-9292/10/10/1163>.
4. Zengwei Yao, Zihao Wang, Weihuang Liu, Yaqian Liu, Jiahui "Panemotion recognition using fusion of three multi-task learning-based classifiers: HSF-DNN, MS-CNN and LLD-RNN". *Science Direct*. [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S0167639319300329>.

Автоматизація оцінки ризиків стартапів за допомогою методів штучного інтелекту

*Дзядевич Дмитро Дмитрович,
студент групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
dimadz2002@gmail.com*

*Науковий керівник: Садовенко Володимир Сергійович,
кандидат фізико-математических наук, доцент
професор кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Стартапи, працюючи в умовах високої конкуренції та невизначеності, потребують ефективних інструментів управління ризиками. Традиційні методи аналізу ризиків зазвичай є трудомісткими та не адаптованими до змінного середовища. Методи штучного інтелекту (ШІ) дозволяють автоматизувати процес оцінки ризиків, підвищуючи швидкість, точність і об'єктивність аналізу.

Постановка задачі.

Розробка та впровадження систем автоматизованої оцінки ризиків стартапів на основі методів штучного інтелекту, що дозволить скоротити витрати ресурсів та підвищити точність управлінських рішень.

Мета дослідження.

Інтеграція методів штучного інтелекту для автоматизації аналізу ризиків у стартапах з метою зниження їх впливу на ключові бізнес-процеси.

Результати дослідження.

Методи ШІ, такі як дерева рішень, логістична регресія та нейронні мережі, дозволяють будувати прогнозні моделі, які враховують широкий спектр ризиків.

Основні етапи автоматизації:

1. Збір даних. Використання внутрішніх та зовнішніх джерел даних, таких як фінансові показники, ринкові тренди та поведінкові дані клієнтів.

2. Моделювання ризиків. Алгоритми ШІ дозволяють класифікувати ризики за рівнями небезпеки (високий, середній, низький).
3. Прогнозування. Використання моделей регресії для оцінки ймовірності виникнення ризиків.
4. Інтеграція в процеси. Результати моделей інтегруються у системи управління проектами стартапу.

Приклад: у фінансових стартапах автоматизація ризик-менеджменту за допомогою нейронних мереж дозволила скоротити ризики кредитних дефолтів на 25% та зменшити час на оцінку позичальників на 40%. У логістичних компаніях застосування алгоритмів кластеризації допомогло знизити затримки постачання на 20%.

Висновки та перспективи.

Методи ШІ є важливим інструментом для автоматизації оцінки ризиків стартапів. Подальший розвиток включає адаптацію алгоритмів до специфічних галузей та інтеграцію з іншими бізнес-інструментами для комплексного управління ризиками.

Список використаних джерел:

1. Савченко О., Прокопенко Є. "Методика оцінки ризиків", Збірник наукових праць, 2018.
2. JSON Web Token Introduction - jwt.io – [Електронний ресурс]. Режим доступу: <https://jwt.io/introduction>.
3. Hinton G. "Artificial Intelligence in Modern Systems", Journal of AI Research, 2020.

Застосування шаблонів проєктування Factory та Observer у рушії Unity для створення інвентарю гравця

*Бойко Микита Сергійович,
студент групи ПДМ-63
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
boiko.m327@gmail.com
Науковий керівник: Бондарчук Андрій Петрович
доктор технічних наук, професор,
професор кафедри Інформаційних систем та технологій
Державного університету інформаційно-комунікаційних технологій*

У цій тезі буде розглянуто використання шаблону проєктування "Factory" й "Observer" на ігровому двигуні Unity для обробки взаємодії між об'єктами інвентаря та гравцем. Factory використовується для створення об'єктів без необхідності знати їх типи. В ігровому інвентарю він може бути використаний для створення різних типів предметів або ресурсів, не прив'язуючись до їх конкретних класів.

Шаблони проєктування "Factory" виник у світі програмування досить давно і став одним з основних шаблонів створення об'єктів. Вперше про нього згадують у книзі "Design Patterns: Elements of Reusable Object-Oriented Software", яка була видана в 1994 році. Ця книга написана чотирма авторами, які отримали прізвище "Бандо четвірка" (Gang of Four): Еріхом Гаммой (Erich Gamma), Річардом Хелмом (Richard Helm), Ральфом Джонсоном (Ralph Johnson) та Джоном Вліссідесом (John Vlissides).

Вони сформулювали шаблон "Factory" як спосіб створення об'єктів, при якому вся логіка створення об'єкта знаходиться у відповідній фабриці, а клієнтський код отримує готовий екземпляр, не знаючи конкретних деталей процесу створення.

Відтоді шаблон "Factory" став популярним і широко використовуваним у програмуванні, включаючи розробку ігор на Unity та інші платформи.

У наступному прикладі використовується Factory, завдяки якому створюється крос-платформні елементи інтерфейсу, які не прив'язують основний код програми до конкретних класів елементів.

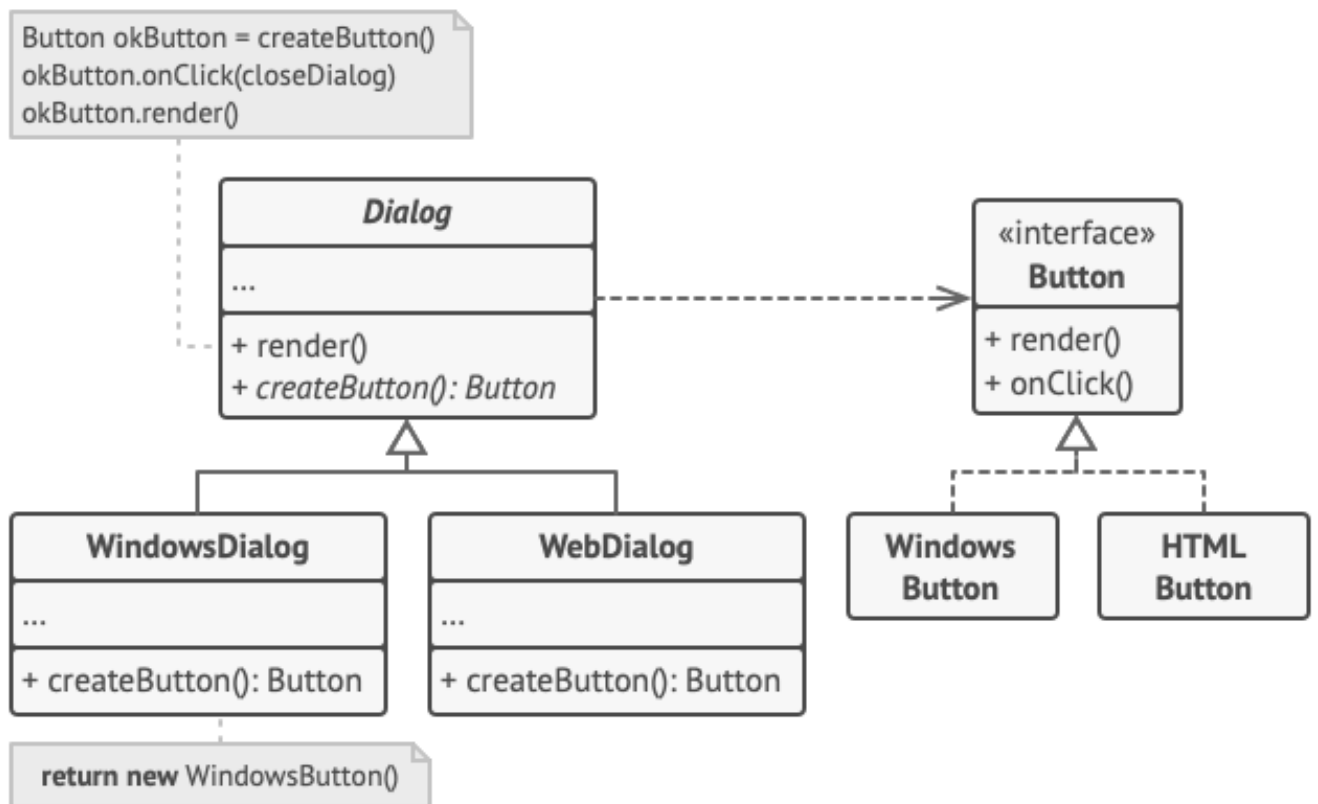


Рис.1. Діаграма класів крос-платформного діалогу

Шаблон проектування "Observer" також виник у тому ж самому джерелі, що і "Factory", а саме у книзі "Design Patterns: Elements of Reusable Object-Oriented Software". Він реалізує механізм підписки, де об'єкт, який є наглядцем, автоматично сповіщає своїх спостерігачів про будь-які зміни у своїх внутрішніх станах. Цей шаблон дозволяє створювати зв'язки "один до багатьох" між об'єктами, де один об'єкт (наглядач) автоматично повідомляє про зміни всім своїм спостерігачам.

У контексті створення ігрового інвентарю на Unity, шаблон "Observer" може бути використаний для сповіщення про зміни вмісту інвентарю, таких як додавання

нових предметів, видалення чи зміна статусу предметів тощо. Наприклад, коли гравець знаходить або додає новий предмет до свого інвентарю, об'єкт інвентарю може повідомити всіх спостерігачів (наприклад, інтерфейс користувача, систему відображення тощо) про цю зміну. Таким чином, інші частини програми можуть оновити свій стан або відобразити новий предмет в інвентарі користувача.

Використання цього шаблону для ігрового інвентарю дозволяє забезпечити розділення логіки відображення предметів в інтерфейсі від логіки їх зберігання та маніпулювання, що робить код більш гнучким і легше піддається змінам.

У цьому прикладі "Observer" дозволяє об'єкту текстового редактора сповіщати інші об'єкти про зміни свого стану.

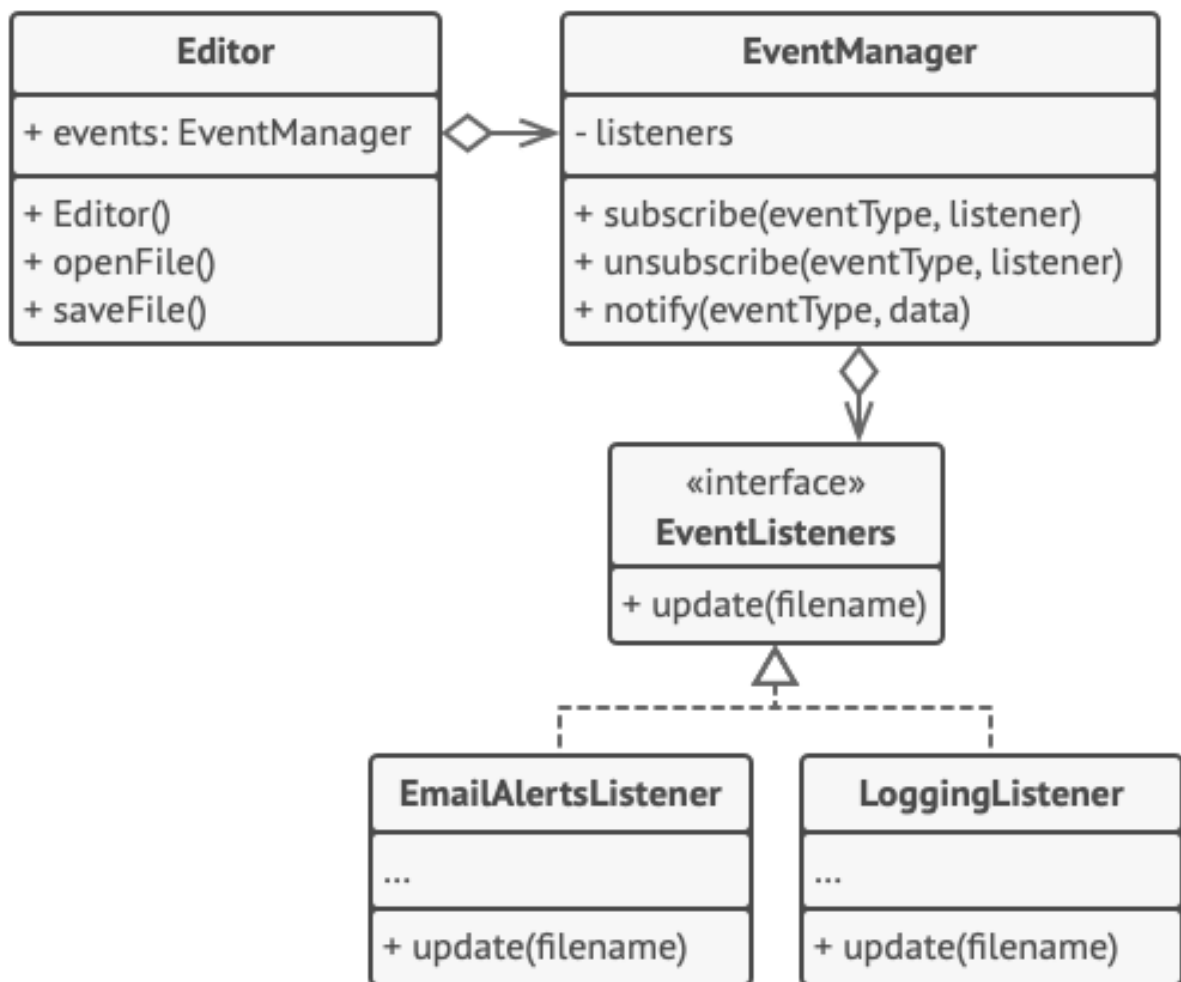


Рис. 2. Діаграма класів оповіщення об'єктів про події в інших об'єктах

Загалом, шаблони проектування "Observer" й "Factory" є потужними інструментами для покращення організації та розширення функціональності програмного забезпечення, включаючи ігрові додатки на Unity. Обидва ці шаблони можуть використовуватися окремо або в поєднанні між собою та іншими шаблонами для створення добре організованого, гнучкого та розширюваного програмного забезпечення. Вони є важливою частиною інструментарію будь-якого розробника програмного забезпечення, включаючи тих, хто працює над ігровими додатками на Unity.

Список використаних джерел:

1. Refactoring Guru [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: <https://refactoring.guru/uk/design-patterns>.
2. Gang of Four Design Patterns [Електронний ресурс] // scholarhat. – 2024. – Режим доступу до ресурсу: <https://www.scholarhat.com/tutorial/designpatterns/gang-of-four-gof-design-patterns>

Застосування методів процедурної генерації рівнів для ігор жанру Tower Defence

*Косенко Денис Максимович,
студент групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
denhelper90@gmail.com
Науковий керівник: Гребенюк Віктор Вікторович,
доктор філософії (PhD),
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Сучасні відеоігри потребують постійного оновлення контенту та створення унікальних рівнів для забезпечення різноманітного ігрового досвіду. Ігри Tower Defense, де стратегічний компонент є ключовим елементом, особливо вимагають динамічних ігрових карток, які надають гравцям нові виклики. Процедурна генерація,

зокрема використання шумових функцій є перспективним підходом до автоматизації створення таких рівнів. Розробка інструменту, що дозволяє автоматично генерувати карти та розміщувати об'єкти навколишнього середовища, дозволить оптимізувати процес розробки та покращити якість контенту. [1-2].

Постановка задачі.

Завданням дослідження є розробка методу процедурної генерації для ігор жанру Tower Defence з використанням шумових функцій та алгоритмів розміщення об'єктів оточення.

Мета дослідження.

Визначення ключових вимог до процедурної генерації карт і методів їх реалізації для автоматичного створення ігрових рівнів з урахуванням особливостей жанру Tower Defence.

Результати дослідження.

Для розробки програмного забезпечення було реалізовано метод процедурної генерації, заснований на використанні алгоритмів шуму Перліна та модульної генерації. Це дозволило створити інструмент для автоматизованої генерації ігрових карток, що відповідають параметрам ігрового світу у жанрі Tower Defense. Основна увага була приділена на впровадження адаптивного підходу до розташування об'єктів, таких як дерева, каміння та інші елементи ландшафту, що забезпечує реалістичність та варіативність карт.

Система підтримує розподіл об'єктів за колірною картою текстури, яка виступає як вхідний шаблон для генерації рівня. Крім того, розроблено механізм визначення областей зіткнень, що дозволяє уникнути перекриття об'єктів. Розроблене програмне забезпечення також містить інтерактивний інтерфейс для тестування ігрових карт та внесення змін до параметрів генерації, що значно спрощує процес перевірки та виправлення створених рівнів.

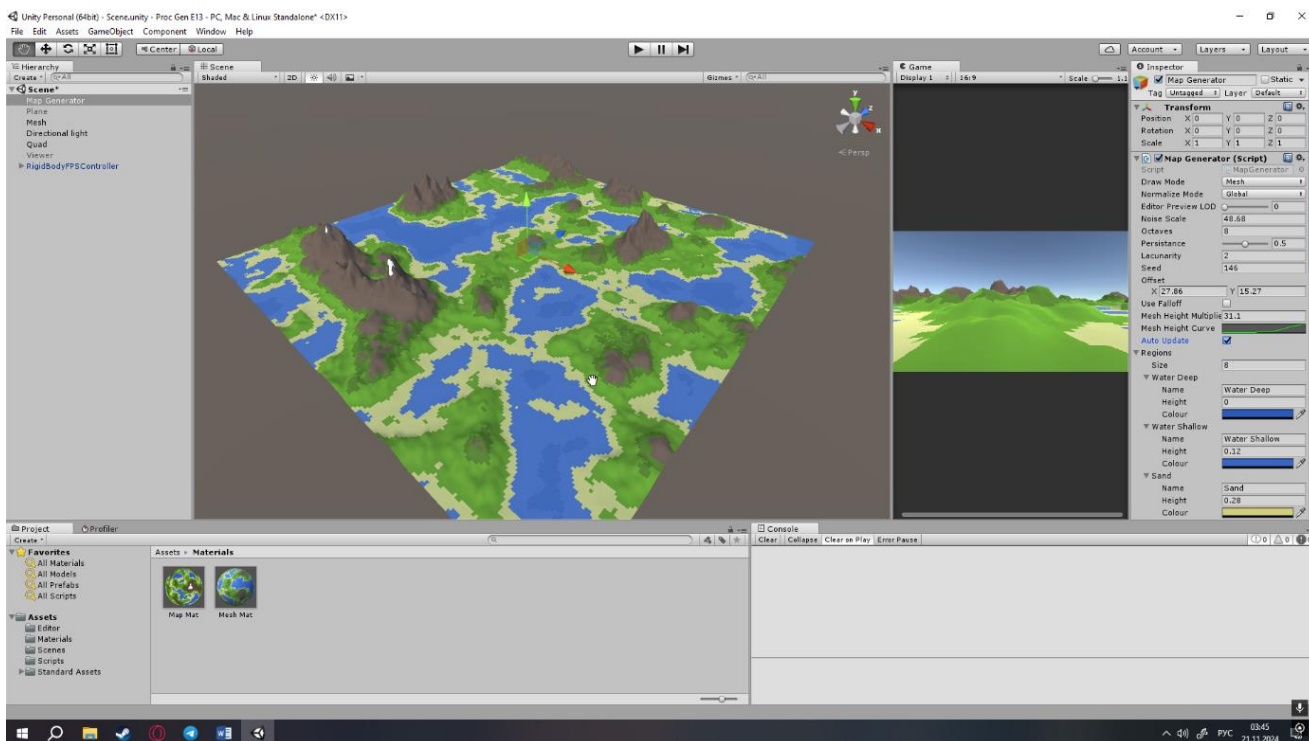


Рис 1. Процедурно згенерована мапа

Цю методологію можна масштабувати до створення карт інших жанрів, де генерація оточення грає ключову роль. Надалі розробка може бути інтегрована в існуючі ігрові двигуни або послужити основою нових проектів в області відеоігор [2-4].

Висновки та перспективи.

Розроблений метод процедурної генерації дозволяє автоматизувати створення ігрових рівнів, що суттєво скорочує тимчасові та ресурсні витрати при розробці ігор жанру Tower Defence. Отримані результати можуть бути використані при розробці нових ігор, забезпечуючи динаміку ігрового процесу та підвищуючи реґрабельність продукту. Подальші дослідження можуть бути спрямовані на розширення функціоналу генерації, зокрема інтеграцію алгоритмів генерації маршрутів ворогів та інших ігрових механік, що дозволить створити повноцінну екосистему для рівнів у Tower Defence.

Список використаних джерел:

1. Notes on Procedural Map Generation Techniques – Christian Mills. Christian Mills. [Електронний ресурс] – Режим доступу : <https://christianjmill.com/posts/procedural-map-generation-techniques-notes/>.
2. Cox G., Piwowarek G. Procedural Generation of Computer Game Maps. Baeldung. [Електронний ресурс] – Режим доступу : <https://baeldung.com/cs/gameplay-maps-procedural-generation>.
3. Дібрівний О.А., Гребенюк В.В. Вступ до об'єктно орієнтованого програмування C#: Навчальний посібник. – К.: Державний університет телекомунікацій, 2018, 190 с.

Роль штучного інтелекту у розвитку сучасних відеоігор

*Левчук Микола Петрович,
студент групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
levchuk036@gmail.com*

*Науковий керівник: Шахматов Іван Олександрович,
викладач кафедри Інформаційних систем та технологій
Державного університету інформаційно-комунікаційних технологій*

У сучасній ігровій індустрії штучний інтелект стає невід'ємним елементом, який визначає розвиток відеоігор та підвищує рівень їх інтерактивності й реалістичності. Завдяки ШІ, гравці отримують можливість взаємодіяти з більш реалістичними персонажами, спостерігати складні стратегії ворогів та занурюватися у динамічні ігрові сценарії. ШІ використовується для створення адаптивних систем, які реагують на дії гравців, роблячи ігровий процес унікальним і захоплюючим. Сучасні підходи до впровадження ШІ у відеоігри не лише покращують якість геймплею, але й сприяють постійному розвитку ігрових технологій та індустрії загалом..

Постановка задачі

Дослідження спрямоване на аналіз ролі ШІ у сучасних відеоіграх, зокрема на оцінку його впливу на розвиток ігрових систем, а також дослідження потенціалу вдосконалення геймплею через його ефективне використання.

Мета дослідження

Метою дослідження є оцінка поточних методів застосування ШІ в сучасних відеоіграх, а також аналіз того, як різні підходи до ШІ дозволяють розробникам створювати більш реалістичні та захоплюючі ігрові світи. Дослідження спрямоване на виявлення ключових тенденцій розвитку ШІ в ігровій індустрії та його потенціалу для подальшого вдосконалення ігор.

Результати дослідження

1. Проведено аналіз різних моделей ШІ, таких як скінченні автомати, машинне навчання та нейронні мережі, що використовуються в сучасних відеоіграх.
2. Виявлено ключові переваги ШІ у покращенні реалістичності поведінки NPC (неконтрольованих персонажів), що робить ігровий досвід більш захоплюючим.
3. Продемонстровано, як ШІ дозволяє динамічно змінювати складність гри відповідно до рівня гравця, забезпечуючи унікальні та персоналізовані сценарії.
4. Розглянуто вплив ШІ на багатокористувацькі ігри, де він може створювати реалістичних опонентів, коли не вистачає реальних гравців.

Висновки та перспективи.

Штучний інтелект став одним із ключових рушіїв інновацій у сучасній ігровій індустрії, впливаючи на всі аспекти ігрового процесу — від поведінки NPC (неконтрольованих персонажів) до складності геймплею і загальної взаємодії гравця з ігровим середовищем. Сучасні методи інтеграції ШІ дозволяють створювати персонажів, здатних навчатися, адаптуватися та демонструвати складні поведінкові моделі, що суттєво підвищує рівень занурення та інтересу до гри. Використання таких підходів, як поведінкові дерева, скінченні автомати та алгоритми машинного

навчання, робить штучний інтелект важливим інструментом для вдосконалення геймплею та підвищення задоволення гравців.

Однак, з розвитком ігрових технологій, з'являються й нові виклики. Сучасні ігри стають більш масштабними, що вимагає ефективніших і оптимізованих рішень для реалізації ШІ. Одним із основних напрямків розвитку є подальша оптимізація алгоритмів ШІ, щоб забезпечити їхню продуктивність на різних платформах, включаючи мобільні пристрої. Важливим також є питання збалансованості ігрового процесу, щоб ШІ був викликом для гравців, але не ставав занадто передбачуваним або надто складним.

Таким чином, роль ШІ в сучасних відеоіграх є надзвичайно важливою і продовжує зростати. Майбутній розвиток цієї технології забезпечить ще більш інтерактивний і реалістичний ігровий досвід, який залучатиме гравців на глибшому рівні. Впровадження нових моделей і методів ШІ в поєднанні з апаратними інноваціями може кардинально змінити підхід до створення відеоігор у найближчі роки, роблячи їх ще більш технологічно складними, багатограними та захоплюючими для різних аудиторій.

Список використаних джерел:

1. What Is AI in Gaming? – [Електронний ресурс] – Режим доступу: <https://www.arm.com/glossary/ai-in-gaming> (дата звернення: 28.09.2024).
2. What is AI in Gaming Industry – [Електронний ресурс] – Режим доступу: <https://www.engati.com/blog/ai-for-gaming> (дата звернення: 01.10.2024).
3. How AI in Gaming Propelling the Industry into a New Epoch – [Електронний ресурс] – Режим доступу: <https://appinventiv.com/blog/ai-in-gaming> (дата звернення: 20.10.2024).
4. How the computer games industry is embracing AI – [Електронний ресурс] – Режим доступу: <https://www.bbc.com/news/business-68844761> (дата звернення: 17.10.2024).

Визначення вимог до організації роботи додатку з користувачем у контексті доповненої реальності

*Линник Ярослав Андрійович,
студент групи ПДМ-61
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
lynyuk.yaroslav@gmail.com
Науковий керівник: Довженко Тимур Павлович,
доцент кафедри, кандидат технічних наук
Державного університету інформаційно-комунікаційних технологій*

Доповнена реальність (AR) швидко розвивається, що створює попит на більш інтуїтивні, функціональні та безпечні способи взаємодії з користувачем. Однією з основних задач дослідження є розробка вимог до AR-додатків, що зможуть покращити користувацький досвід, зменшити когнітивне навантаження та забезпечити доступність для різних категорій користувачів [1, 2].

Постановка задачі.

Доповнена реальність (AR) є одним із перспективних напрямків сучасних технологій, що вимагає адаптації інтерфейсу додатків до специфічних потреб користувачів. Основною задачею цього дослідження є визначення ключових вимог до взаємодії з користувачем, враховуючи особливості AR, такі як інтеграція об'єктів у реальне середовище та забезпечення зручності роботи [2, 3].

Мета дослідження.

Метою є визначення ключових вимог до інтерфейсу AR-додатків, які забезпечують природну і зручну взаємодію користувача з віртуальними об'єктами, інтегрованими в реальне середовище [4].

Результати дослідження.

У ході роботи визначено кілька основних принципів:

- забезпечення інтуїтивності інтерфейсу завдяки мінімалістичному дизайну, що сприяє швидкому засвоєнню [1];

- впровадження нових підходів до взаємодії з AR, таких як інтеграція тактильних посередників, що дозволяє користувачам більш ефективно взаємодіяти з віртуальними об'єктами [2];
- оптимізація продуктивності додатку з урахуванням обмежень мобільних пристроїв, таких як автономність батареї та продуктивність GPU [3];
- адаптація додатку до різних сценаріїв використання, наприклад, навігація, навчання або розваги [4].

Побудовано прототип інтерфейсу (рисунок 1), який дозволяє користувачам взаємодіяти з об'єктами в AR середовищі.

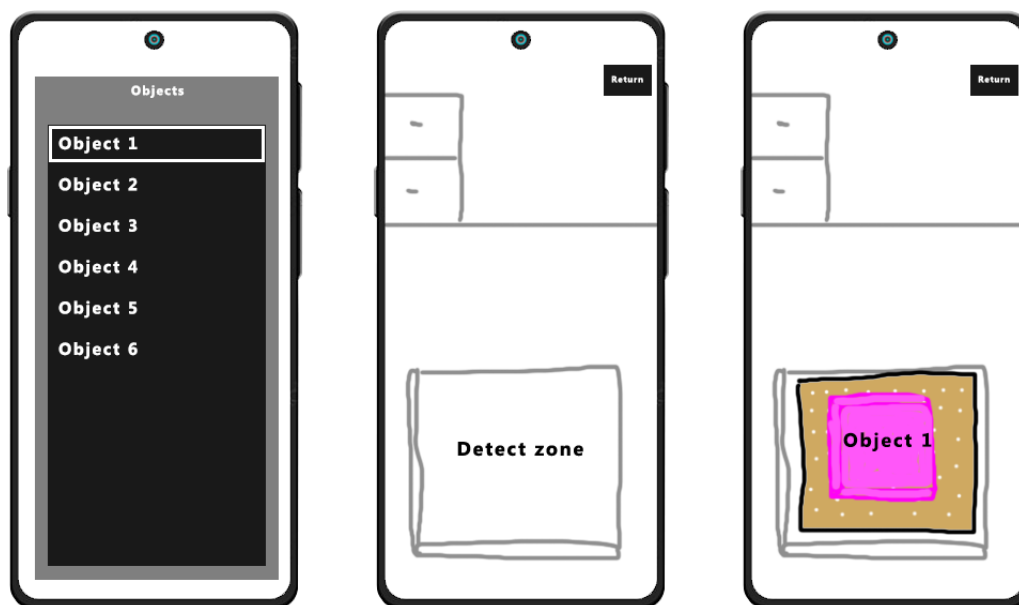


Рис. 1 Прототип інтерфейсу AR-додатку

Висновки та перспективи.

Розроблені вимоги можуть стати базою для створення сучасних AR-додатків, які відповідають потребам широкої аудиторії. Подальше дослідження буде спрямоване на тестування прототипу у реальних умовах та адаптацію під різні платформи [1, 4].

Список використаних джерел:

1. Azuma, R. "A Survey of Augmented Reality." Presence: Teleoperators and Virtual Environments. [Електронний ресурс] Режим доступу: https://doi.org/10.1162/pres_a_00340
2. Yue Li, Eugene Ch'ng, Sue Cobb, Simon See. "Presence and Communication in Hybrid Virtual and Augmented Reality Environments." [Електронний ресурс] Режим доступу: https://doi.org/10.1162/pres_a_00350
3. Maximino Bessa, Miguel Melo, Aliane Loureiro Krassmann. "What Is the Relationship between the Sense of Presence and Learning in Virtual Reality?" [Електронний ресурс] Режим доступу: https://doi.org/10.1162/pres_a_00352
4. Unity Technologies. "Augmented Reality Best Practices." [Електронний ресурс]. Режим доступу: <https://learn.unity.com/tutorial/ar-best-practices>

User Interface Generation using AI technology

*Mykolaienko Vladyslav Oleksandrovich,
student of SE-32,
specialty 121 Software engineering,
State University of Information and Communication Technologies
mykolaienkovlad@gmail.com
Academic supervisor: Storchak Kamila Pavlivna,
Doctor of technical sciences, professor,
Head of the Department of Information Systems and Technologies
State University of Information and Communication Technologies*

User interfaces (UIs) are a key part of software, affecting how users interact with digital products. Creating effective UIs has traditionally been a complex and time-consuming task, requiring both design and technical skills. However, with the rise of Artificial Intelligence (AI), generating UIs can now be faster and more efficient. AI technology can analyze data, learn from existing designs, and create user-friendly interfaces

automatically. This allows developers to focus more on functionality while AI handles the design. Recent advancements, like machine learning and generative models, help AI systems make smart design choices that adapt to different user needs and devices. This thesis explores how AI can be used to generate UIs, the advantages it offers, and the challenges that come with it. It aims to show how AI can simplify the design process, make it faster, and improve the user experience.

Problem Statement.

Creating user interfaces is a time-consuming process that requires balancing good design and functionality. Traditional methods rely on manual work by designers and developers, leading to slow development and inconsistent results. Adapting UIs for different devices adds even more complexity. There is a need for smarter tools that can automate the UI design process while still delivering quality results. AI technology offers a solution, but integrating it effectively into UI design is challenging. This thesis aims to explore how AI can simplify and speed up UI creation while maintaining a high standard of usability.

Research Objective.

The objective of this research is to investigate how AI technology can be effectively used to automate and enhance the generation of user interfaces. Specifically, this study aims to:

1. Explore the potential of AI in automating UI design to reduce development time and effort.
2. Evaluate the quality, consistency, and usability of AI-generated interfaces compared to traditional methods.
3. Identify the challenges and limitations of integrating AI in UI design.
4. Propose strategies and best practices for successfully incorporating AI tools in the UI development workflow.

This research seeks to demonstrate that AI can streamline the UI design process, making it faster, more adaptive, and more user-friendly.

Research Results.

Creating AI powered UI it's not an easy task , and for today current hardware on the phones can give us the opportunity to do that, and in order to do that we need to pass a few stages .

First stage its create a database for AI , this database will include

1.Page structures - each page on your phone has some goals so, each goal needs a different page structure.

2.Components - each page must include as minimum 1 component, this component must be optimized for different screens and scenarios.

3. User behavior data - very important to understand how users use pages,components, and how they interact with them , we will collect this data for future analysis

Second stage it's provided to the user first screen with one input filed wherein he will write how he wants to interact with your business app, and transform this prompt into user behavior app file , this files will be generated by Large Language Model , after we will get it - we can generate user pages depending on his favors and provided the best user experience.

After we have collected this we need to generate an intermediate description of our components, their name's, and main reason.

```
pages:
- page:
  name: "{{ page_name1 }}" # Specific page name
  description: "{{ page_description1 }}" # Main purpose of the specific page
  components:
  - name: "{{ specific_component1_name }}" # Name of a specific component for this pa
    description: "{{ specific_component1_description }}" # Main purpose of the specif
  - name: "{{ specific_component2_name }}"
    description: "{{ specific_component2_description }}"

- page:
  name: "{{ page_name2 }}" # Another specific page name
  description: "{{ page_description2 }}" # Main purpose of this specific page
  components:
  - name: "{{ specific_component3_name }}" # Name of another specific component
    description: "{{ specific_component3_description }}" # Main purpose of this speci
  - name: "{{ specific_component4_name }}"
    description: "{{ specific_component4_description }}"
```

Picture 1 (Generated Intermediate Description)

After Generated Intermediate Description was created we can pass each our top level representation of page and components to the LLM and create from this ready to use widget.

Conclusions and Future Prospects.

AI technology has the potential to revolutionize user interface design by automating and speeding up the process. This research shows that AI can create effective and adaptable UIs, saving time and reducing manual work. However, challenges remain in ensuring AI-generated designs meet high-quality standards and adapt to diverse user needs. In the future, improvements in AI models could lead to even smarter and more responsive UIs. Further research could explore more advanced AI tools, better integration with existing design systems, and ways to make AI-generated interfaces more personalized and intuitive.

References:

1. M. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016, pp. 123-145.
2. K. Schwab, "The Fourth Industrial Revolution," *World Economic Forum*, 2017, pp. 78-92.
3. J. Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed., Cambridge University Press, 2009, pp. 230-255.
4. P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, Basic Books, 2015, pp. 101-115.
5. Apple Inc., *Human Interface Guidelines*, Apple Developer, 2021. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/>.
6. Google LLC, *Material Design Guidelines*, Google Design, 2022. [Online]. Available: <https://material.io/design/>.
7. E. Siegel, *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*, Revised and Updated ed., Wiley, 2016, pp. 200-220.
8. D. Norman, *The Design of Everyday Things*, Revised and Expanded ed., Basic Books, 2013, pp. 150-170.
9. A. McKinney, *Artificial Intelligence in UX Design: A Design Approach*, Apress, 2020, pp. 45-60.
10. B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 6th ed., Pearson, 2016, pp. 95-110.

Інтеграція сучасних інформаційних технологій у процес UX-оптимізації веб-платформ ігрових спільнот

Юхта Максим Артемович

студент групи ПДМ-63

спеціальність 121 Інженерія програмного забезпечення

Державного університету інформаційно-комунікаційних технологій

BezNika2012@gmail.com

Науковий керівник: Компан Сергій Володимирович

кандидат фізико-математичних наук

доцент кафедри Інтернет-технологій

Державного університету інформаційно-комунікаційних технологій

Постановка задачі.

Сучасна ігрова індустрія активно розвивається у цифровому середовищі, а веб-платформи ігрових спільнот відіграють важливу роль у залученні, утриманні та підтримці гравців. Перед розробниками постає завдання створення інтуїтивних і функціональних веб-платформ, які надають персоналізований досвід. Інтеграція сучасних інформаційних технологій, таких як штучний інтелект (ШІ), аналіз великих даних (Big Data) та алгоритми машинного навчання (ML), є ключовим інструментом у досягненні цієї мети.

Мета дослідження.

Визначення можливостей і переваг впровадження інформаційних технологій для оптимізації користувацького досвіду (UX) веб-платформ ігрових спільнот, а також оцінка їхнього впливу на ключові показники ефективності: час перебування на платформі, залученість користувачів і коефіцієнт конверсії.

Результати дослідження.

Популярне використання ШІ та Big Data дозволяє персоналізувати платформу для кожного користувача за допомогою аналізу великих обсягів даних, включаючи дії користувачів, їхні уподобання та історію взаємодії з платформою.

Системи ШІ можуть:

генерувати рекомендації, наприклад, пропонувати популярні чи нові ігри залежно від вподобань користувача;

аналізувати поведінку користувачів у реальному часі для динамічного налаштування контенту на головній сторінці;

виявляти потенційні труднощі, наприклад, високий рівень відмов чи зменшення часу перебування на платформі, і пропонувати рішення.

Алгоритми Big Data допомагають оптимізувати роботу платформ, забезпечуючи безперебійну роботу серверів навіть під час пікових навантажень, наприклад, у дні запуску великих ігрових подій. Машинне навчання (ML) відкриває нові горизонти для UX-оптимізації: Прогнозування поведінки користувачів дозволяє адаптувати інтерфейс відповідно до їхніх уподобань; Оптимізація дизайну здійснюється через автоматичне тестування різних версій інтерфейсу (A/B тестування); Інтеграція систем розпізнавання мовлення та тексту підвищує інтерактивність платформи.

Для залучення користувачів платформи використовуються динамічні візуалізації, 3D-графіка та гейміфікація, що дозволяє створити інтерактивний досвід. Виклики впровадження. Інтеграція сучасних технологій має свої виклики: Значні фінансові витрати на розробку та підтримку; Складність масштабування для обробки великих обсягів запитів від багатьох користувачів одночасно; Забезпечення безпеки даних і дотримання конфіденційності користувачів;

Висновки та перспективи.

Дослідження підтверджують, що використання сучасних інформаційних технологій для UX-оптимізації підвищує ефективність платформ. Наприклад, впровадження системи персоналізації на основі ML збільшило час взаємодії користувачів із платформою в середньому на 25%.

Подальші дослідження включають розвиток гібридних моделей, які об'єднують можливості кількох мов програмування для створення більш адаптивних і багатофункціональних рішень.

Список використаних джерел:

1. Web Development in the Gaming Industry: Creating Immersive Experiences | MoldStud. Custom software development company | MoldStud. – [Електронний ресурс] – Режим доступу: <https://moldstud.com/articles/p-web-development-in-the-gaming-industry-creating-immersive-experiences> (дата звернення: 20.11.2024).
2. Enhancing User Experience: Key Tips for Gaming Platforms. UI/UX Design Services Firm - UX Specialist in WANDR. – [Електронний ресурс] – Режим доступу: <https://www.wandr.studio/blog/user-experience-for-gaming-platforms-best-practices> (дата звернення: 24.11.2024).
3. Agner L., Neczyk B., Renzi A. Recommendation Systems and Machine Learning: Mapping the User Experience. SpringerLink. – [Електронний ресурс] – Режим доступу: https://link.springer.com/chapter/10.1007/978-3-030-49760-6_1 (дата звернення: 24.11.2024).

Використання LOD-моделей адаптивного освітлення для оптимізації рендерингу

*Яковчук Василь Андрійович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
vasyakov4uk@gmail.com
Науковий керівник: Залива Віталій Вікторович,
доктор філософії (PhD),
ст. викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Підвищення графічних можливостей в ігрових двигунах залишається однією з ключових задач сучасної розробки ігор. Адаптивне освітлення є ключовим фактором, що впливає на продуктивність, особливо у великих відкритих сценах. Зокрема, це стосується оптимізації освітлення, яка забезпечує баланс між якістю візуалізації та продуктивністю системи.

Постановка задачі.

Дослідити метод, що дозволяє адаптивно змінювати параметри освітлення залежно від положення об'єкта від гравця (див. рис. 1), використовуючи принцип рівнів деталізації (LOD).



Рис 1. Перехід між рівнями деталізації тіні

Мета дослідження.

Підвищення продуктивності графічного рендерингу за рахунок адаптації принципу (LOD), який детально описаний у роботі Luebke et al. [1] до джерел освітлення.

Результати дослідження.

Адаптивний метод шейдингу освітлення з використанням LOD-моделей, дозволяє застосовувати високодеталізовані налаштування освітлення лише для об'єктів у зоні фокусу гравця. Для об'єктів, що покидають цю зону, деталізація освітлення автоматично зменшується. Цей підхід узгоджується з ідеями Clarberg et al. [2].

Висновки та перспективи.

Метод показав ефективне використання графічних ресурсів у реальному часі та збільшення числа кадрів за секунду до 20% в залежності від налаштувань та несуттєво впливаючи на якість фінального кадру. Подальший розвиток методу може включати його поєднання з техніками трасування променів, що дозволить оптимізувати обчислення глобального освітлення у реальному часі.

Список використаних джерел:

1. Luebke, D., Reddy, M., Cohen, J. D., Varshney, A., Watson, B., & Huebner, R. *Level of Detail for 3D Graphics*, 2003, С. 145.
2. Clarberg, P., Toth, R., Hasselgren, J., & Akenine-Möller, T. *Compressed Multiresolution Hierarchies for Efficient Global Illumination*, 2010, С. 12.

Динамічне прийняття рішень на основі особистісних рис ігрових персонажів

Ярошевський Олександр Вікторович,
студент групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
jaroschew@gmail.com
Науковий керівник: Замрій Ірина Вікторівна,
доктор технічних наук, доцент,
завідувач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій

Сучасні відеоігри стають все більш складними та інтерактивними, прагнучи створити для користувача переконливий досвід занурення. У зв'язку з цим зростає попит на інтелектуальні системи прийняття рішень, що враховують особистісні риси ігрових персонажів, дозволяючи їм діяти з урахуванням своїх характеристик та адаптуватись до змінних умов ігрового середовища. Серед найбільш поширених моделей, що описують риси особистості, можна виділити Big Five, HEXACO та Dark Triad, які забезпечують теоретичну основу для створення алгоритмів особистісного поведінкового моделювання.

Розробка методів динамічного прийняття рішень на основі цих моделей дозволяє глибше інтегрувати особистісні характеристики у процес адаптивної поведінки ігрових персонажів [1].

Постановка задачі

Основним завданням цього дослідження є розробка методів для моделювання динамічного прийняття рішень ігровими персонажами на основі їхніх особистісних характеристик, що включає аналіз теоретичних моделей особистості (Big Five, HEXACO, Dark Triad) для виділення ключових рис, які впливають на прийняття рішень та створення алгоритмів, що дозволяють персонажам адаптувати поведінку до контексту ігрового середовища.

Мета дослідження

Метою роботи є розробка і впровадження методів динамічного прийняття рішень для ігрових персонажів, які дозволять персонажам приймати рішення з урахуванням особистісних характеристик, адаптуючи свою поведінку до різних сценаріїв у грі. Ця мета включає підвищення реалістичності взаємодії користувача з ігровим світом, де персонажі демонструють послідовність та різноманітність у своїх діях.

Результати дослідження

Ідентифікація значущих особистісних рис:

Для Big Five було виділено такі основні риси: відкритість досвіду (T_O), сумлінність (T_C), екстраверсію (T_E), приємність/альтруїзм (T_A), емоційна стабільність (T_N) [2].

HEXACO доповнює Big Five такими характеристиками, як чесність-покірність (T_H), емоційність (T_X) [3].

Dark Triad включає: нарцисизм (T_{Narc}), макіавеллізм ($T_{Machiav}$) та психопатію (T_{Psych}), що характеризують схильність до маніпуляцій, егоцентричність та відсутність емпатії [4].

Аналіз впливу рис на поведінку. Кожна риса може мати різну вагу у процесі прийняття рішень в ігровому середовищі залежно від типу ситуації. Наприклад, для ситуацій соціальної взаємодії доброзичливість та чесність мають більшу вагу, ніж відкритість до досвіду або емоційність [5].

Запропоновано модель, що використовує вагові коефіцієнти для обчислення впливу особистісних рис на результат прийняття рішення. Вона дозволяє кожному персонажу генерувати унікальні рішення залежно від ситуаційних вимог і особистісних характеристик.

Розглядається функція D прийняття рішення, що залежить від особистісних характеристик:

$$D = \frac{\sum_{i=1}^n w_i \cdot T_i}{\sum_{i=1}^n w_i} \cdot 100 ,$$

де T_i – значення особистісної риси персонажа, w_i – вага цієї риси у конкретній ситуації, яка визначається з урахуванням контексту гри, а n – кількість особистісних рис, що враховуються (відповідає характеристикам з Big Five, HEXACO і Dark Triad).

Розподіл ваг для Big Five, HEXACO та Dark Triad:

Big Five: w_O, w_C, w_E, w_A, w_N вагові коефіцієнти для кожної риси.

HEXACO: w_H, w_X – додаткові ваги для характеристик HEXACO.

Dark Triad: $w_{Narc}, w_{Machiav}, w_{Psych}$ – ваги для рис Dark Triad.

В такому випадку значення функції D розраховується наступним чином:

$$D = \frac{w_O \cdot T_O + w_C \cdot T_C + w_E \cdot T_E + w_A \cdot T_A + w_N \cdot T_N + w_H \cdot T_H + w_X \cdot T_X + w_{Narc} \cdot T_{Narc} + w_{Machiav} \cdot T_{Psych}}{w_O + w_C + w_E + w_A + w_N + w_H + w_X + w_{Narc} + w_{Machiav} + w_{Psych}} \cdot 100,$$

Висновки та перспективи

Модель дає змогу точно моделювати рішення персонажа на основі його особистісних характеристик у різних ігрових ситуаціях. Вагові коефіцієнти w_i дозволяють адаптувати вплив конкретних рис під конкретні вимоги ситуацій, забезпечуючи узгоджену і реалістичну поведінку персонажів, що враховує складні взаємозв'язки між різними психологічними характеристиками.

Список використаних джерел:

1. Player behavioural modelling for video games - sciencedirect.com [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/S1875952111000486> (дата звернення: 19.08.2024).
2. Big Five Model of Personality - onlinelibrary.wiley.com [Електронний ресурс] – Режим доступу : <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119547174.ch186> (дата звернення: 19.08.2024).
3. Big Five and HEXACO Personality Traits, Proenvironmental Attitudes, and Behaviors: A Meta-Analysis - journals.sagepub.com [Електронний ресурс] – Режим доступу : <https://journals.sagepub.com/doi/full/10.1177/1745691620903019> (дата звернення: 20.08.2024).
4. A primer on the Dark Triad traits - tandfonline.com [Електронний ресурс] – Режим доступу : <https://www.tandfonline.com/doi/full/10.1111/ajpy.12198#abstract> (дата звернення: 20.08.2024).
5. Player behavior and personality modeling for interactive storytelling in games - sciencedirect.com [Електронний ресурс] – Режим доступу : <https://www.sciencedirect.com/science/article/abs/pii/S1875952118300120?via%3Dihub> (дата звернення: 20.08.2024).

Використання та редагування базового алгоритму псевдовипадкових чисел $C\#$ для генерації подій у грі

*Бойко Микита Сергійович,
студент групи ПДМ-63
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
boiko.m327@gmail.com
Науковий керівник: Бондарчук Андрій Петрович
доктор технічних наук, професор,
професор кафедри Інформаційних систем та технологій
Державного університету інформаційно-комунікаційних технологій*

Створення ігор становить одну з найцікавіших та найшвидше розвиваються сфер у сучасній індустрії розваг. За рахунок досягнень у технологіях та росту

популярності відеоігор, вимоги до створення якісних та захоплюючих ігор зростають щодня.

Генерація псевдовипадкових чисел (ГПВЧ) в іграх є однією з ключових складових, що впливають на різноманітність та непередбачуваність ігрового процесу. Цей процес забезпечує постійний потік нових ситуацій та випробувань для гравців, зробивши гру більш захоплюючою та цікавою. Одним з основних методів генерації псевдовипадкових чисел є використання алгоритмів, таких як лінійний конгруентний метод або метод Фішера-Йетса-Дарсі. Ці алгоритми дозволяють ігровому движку створювати випадкові числа з певними властивостями, такими як рівномірний розподіл, що забезпечує баланс між випадковістю та контрольованістю в грі.

Псевдовипадковість часто виявляється кращою в іграх порівняно з повною випадковістю з кількох причин. По-перше, вона дозволяє забезпечити певний рівень передбачуваності, що важливо для гравців у процесі планування своїх дій та прийняття стратегічних рішень. Гравці можуть адаптувати свою гру, використовуючи знання про псевдовипадкові величини, що робить геймплей більш зграбним і цікавим.

По-друге, псевдовипадковість дозволяє забезпечити більшу різноманітність ігрового досвіду, виключаючи можливість повторення однотипних ситуацій у кожній грі. Це робить гру більш динамічною та захоплюючою, адже гравці завжди стикаються з новими випробуваннями та викликами.

Крім того, використання псевдовипадкових чисел дозволяє ефективно використовувати обмежені ресурси обчислювальної техніки, оскільки генерація повної випадковості може бути дуже ресурсозатратною. Псевдовипадковість дозволяє створювати великий обсяг випадкових даних за короткий час, забезпечуючи швидку та ефективну реалізацію ігрових сценаріїв.

У мові програмування C# одним з основних засобів для генерації псевдовипадкових чисел є клас `Random`, який надає різноманітні методи для отримання випадкових значень. `Random` використовує алгоритм псевдовипадкового генерування, а конкретно лінійний конгруентний метод. Лінійний конгруентний

метод (LCG) базується на рекурсивній формулі, що використовує попереднє значення, щоб згенерувати наступне. Формула виглядає приблизно так: $X_{n+1} = (a \times X_n + c) \bmod m$

X_{n+1} - наступне псевдовипадкове число

X_n - наступне псевдовипадкове число

a , c , i m - константи, які визначаються для кожного конкретного реалізатора LCG.

У класі Random вони вже задані заздалегідь зі значеннями, які забезпечують хороший ступінь випадковості для більшості використовуваних сценаріїв. Однак важливо пам'ятати, що числа, які генерує Random, насправді не є повністю випадковими, оскільки вони залежать від попередніх значень та відомих констант. Ці числа називаються "псевдовипадковими", оскільки вони виглядають випадковими, але насправді є детермінованими. Тому вони називаються псевдовипадковими числами.

Це означає, що якщо ви запустите програму з тими ж самими параметрами і початковим значенням, ви отримаєте ту ж саму послідовність псевдовипадкових чисел. Це може бути корисним для відтворення результатів або відладки програми, але не гарантує повної випадковості в справжньому розумінні цього слова.

Проте, Завдяки відкритому коду класу Random, який можна знайти на репозиторіях GitHub та інших платформах, ви маєте можливість розробити свій власний алгоритм генерації псевдовипадкових чисел, спираючись на існуючі реалізації.

Отже, в рамках даної тези було розглянуто використання алгоритму псевдовипадкових чисел C#. Було проведено огляд літератури та детальний опис алгоритму. Використання алгоритму псевдовипадкових чисел є важливою складовою розробки, яка дозволяє створювати більш складні та цікаві ігрові сценарії.

Список використаних джерел:

1. Random [Електронний ресурс] // .NET Platform.. – 2023. – Режим доступу до ресурсу:
<https://github.com/dotnet/coreclr/blob/ebda0e66df0dd19688cf8a88375bb31c184f5037/src/mscorlib/shared/System/Random.cs>.
2. Кнут Д. Мистецтво програмування / Дональд Ервін Кнут, 2020.

Визначення методів процедурної генерації рівнів для ігор жанру Tower Defence

*Косенко Денис Максимович,
студент групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
denhelper90@gmail.com
Науковий керівник: Гребенюк Віктор Вікторович,
доктор філософії (PhD),
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Сучасні відеоігри потребують постійного оновлення контенту та створення унікальних рівнів для забезпечення різноманітного ігрового досвіду. Ігри Tower Defense, де стратегічний компонент є ключовим елементом, особливо вимагають динамічних ігрових карток, які надають гравцям нові виклики. Процедурна генерація, зокрема використання шумових функцій є перспективним підходом до автоматизації створення таких рівнів. Розробка інструменту, що дозволяє автоматично генерувати карти та розміщувати об'єкти навколишнього середовища, дозволить оптимізувати процес розробки та покращити якість контенту. [1-2].

Постановка задачі.

Завданням дослідження є розробка методу процедурної генерації для ігор жанру Tower Defence з використанням шумових функцій та алгоритмів розміщення об'єктів оточення.

Мета дослідження.

Визначення ключових вимог до процедурної генерації карт і методів їх реалізації для автоматичного створення ігрових рівнів з урахуванням особливостей жанру Tower Defence.

Результати дослідження.

Для розробки методу процедурної генерації використано шум Перліна для створення основи ігрового рівня. Додатково реалізовано алгоритми розміщення об'єктів оточення, які забезпечують унікальність та логічність сформованого середовища. Генерація рівнів виконується за допомогою шумових функцій, які дозволяють створювати різноманітні, реалістичні та збалансовані карти. Процес розміщення об'єктів враховує тип місцевості, створений на основі шумів, та параметри, задані розробником [2-4].

Висновки та перспективи.

Розроблений метод процедурної генерації дозволяє автоматизувати створення ігрових рівнів, що суттєво скорочує тимчасові та ресурсні витрати при розробці ігор жанру Tower Defence. Отримані результати можуть бути використані при розробці нових ігор, забезпечуючи динаміку ігрового процесу та підвищуючи реgrabельність продукту. Подальші дослідження можуть бути спрямовані на розширення функціоналу генерації, зокрема інтеграцію алгоритмів генерації маршрутів ворогів та інших ігрових механік, що дозволить створити повноцінну екосистему для рівнів у Tower Defence.

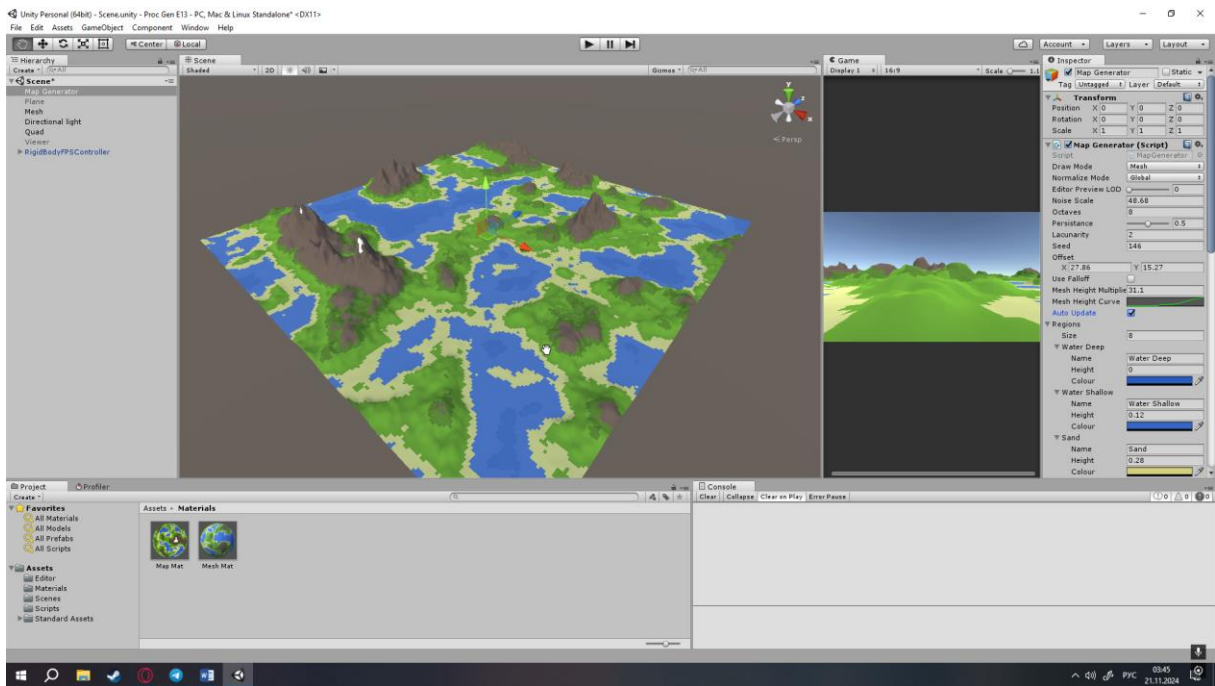


Рис. 1. Процедурно згенерована мапа

Список використаних джерел:

1. Notes on Procedural Map Generation Techniques – Christian Mills. Christian Mills. [Електронний ресурс] – Режим доступу : <https://christianjmills.com/posts/procedural-map-generation-techniques-notes/>.
2. Cox G., Piwowarek G. Procedural Generation of Computer Game Maps. Baeldung. [Електронний ресурс] – Режим доступу : <https://baeldung.com/cs/gameplay-maps-procedural-generation>.
3. Дібрівний О.А., Гребенюк В.В. Вступ до об'єктно орієнтованого програмування C#: Навчальний посібник. – К.: Державний університет телекомунікацій, 2018, 190 с.

Огляд технологій доповненої реальності для візуалізації військової техніки та військових технологій в ігровому рушії unity

*Линник Ярослав Андрійович,
студент групи ПДМ-61
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
lynyuk.yaroslav@gmail.com
Науковий керівник: Довженко Тимур Павлович,
доцент кафедри, кандидат технічних наук
Державного університету інформаційно-комунікаційних технологій*

Кожен день військові фахівці стикаються з необхідністю швидкого та точного оцінювання сучасної військової техніки та технологій. Використання доповненої реальності (AR) для візуалізації військових технологій може значно підвищити ефективність процесу прийняття рішень, навчання та планування операцій. Unity, як потужний рушій для створення AR-додатків, пропонує безліч інструментів для реалізації таких проектів [1].

Постановка задачі.

Провести огляд технологій, що можуть бути використані для візуалізації військової техніки та військових технологій з використанням ігрового рушія Unity.

Мета дослідження.

Дослідити теоретичні та практичні аспекти використання технологій доповненої реальності при візуалізації військової техніки та військових технологій, включаючи їх застосування для навчання, планування операцій, а також оцінки техніко-тактичних характеристик.

Результати дослідження.

Доповнена реальність дозволяє інтегрувати віртуальні об'єкти у реальний світ, забезпечуючи новий рівень взаємодії з інформацією [2]. З використанням Unity,

розробники можуть створювати інтерактивні моделі військової техніки, що допомагають у навчанні, симуляціях бойових дій та стратегічному плануванні.

Основні задачі візуалізації військової техніки за допомогою AR:

- створення детальних 3D-моделей військової техніки;
- інтерактивна демонстрація функціональності та можливостей техніки;
- візуалізація техніко-тактичних характеристик в реальному часі.

Unity надає можливості для інтеграції різних AR-платформ, таких як ARKit, ARCore та Microsoft HoloLens. Це дозволяє створювати кросплатформні AR-додатки, які можна використовувати на різних пристроях, включаючи смартфони, планшети та окуляри доповненої реальності.

Інформаційно-аналітична діяльність у військовій сфері включає збір, аналіз та представлення даних про технічні характеристики та бойові можливості військової техніки [3]. Використання AR для цих цілей може значно покращити точність та швидкість оцінки ситуації.

Створення AR-моделей військової техніки включає кілька етапів:

1. Збір даних та створення 3D-моделей.
2. Інтеграція моделей в AR-додаток за допомогою Unity.
3. Розробка інтерфейсу користувача для взаємодії з моделями.
4. Тестування та оптимізація додатку для різних платформ.

Доповнена реальність також може використовуватись для візуалізації комплексних систем, таких як системи управління військами, логістика та постачання, що дозволяє військовим фахівцям отримувати візуальне уявлення про взаємодію різних компонентів системи в реальному часі. На рисунку 1 можна побачити, як за допомогою мобільного застосунку в реальному світі, через камеру, на стіл було додано віртуальний об'єкт у вигляді шару.

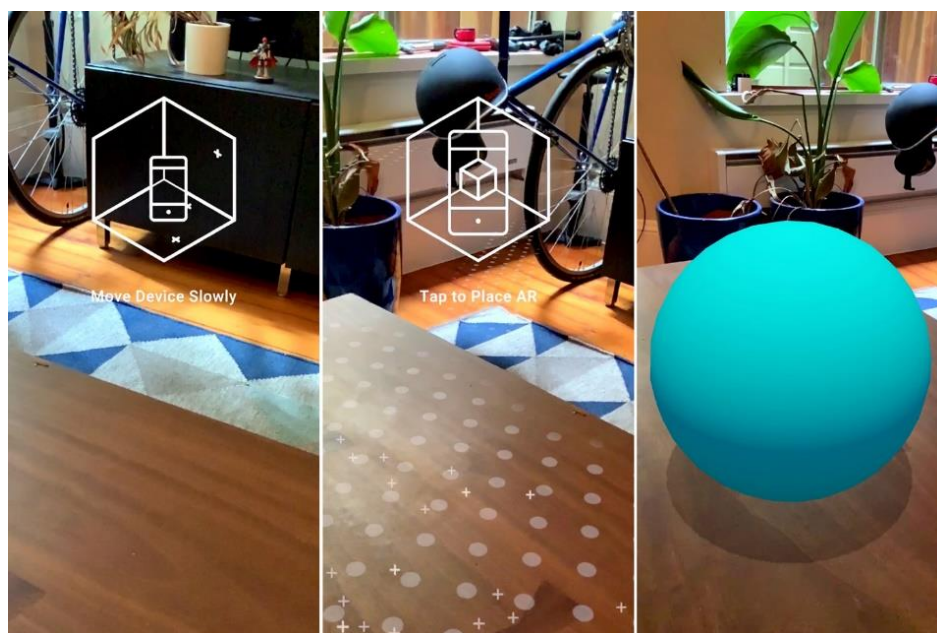


Рис. 1 Доповнення реальності віртуальною кулею

Висновки та перспективи.

Використання технологій доповненої реальності для візуалізації військової техніки та військових технологій з використанням Unity може значно покращити ефективність навчання, планування та прийняття рішень у військовій сфері. Серед перспектив дослідження можна виділити подальший розвиток AR-технологій, інтеграцію їх з іншими інформаційними системами та використання у широкому спектрі військових застосувань.

Список використаних джерел:

1. Unity Technologies. AR Foundation Documentation. Режим доступу: <https://docs.unity3d.com/Packages/com.unity.xr.foundation@latest>
2. Azuma, R. (1997). A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments. Режим доступу: <https://www.cs.unc.edu/~azuma/ARpresence.pdf>
3. Milgram, P., & Kishino, F. (1994). A Taxonomy of Mixed Reality Visual Displays. IEICE Transactions on Information and Systems.

Огляд методів провадження штучного інтелекту в гру на ігровому рушії Unity

*Левчук Микола Петрович,
студент групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
levchuk036@gmail.com*

*Науковий керівник: Шахматов Іван Олександрович,
викладач кафедри Інформаційних систем та технологій
Державного університету інформаційно-комунікаційних технологій*

У сучасних ігрових додатках штучний інтелект (ШІ) відіграє ключову роль у створенні захоплюючих та інтерактивних ігрових середовищ. Використання ШІ в іграх дозволяє створювати реалістичних персонажів, поліпшувати поведінку ворогів та союзників, а також забезпечувати динамічний геймплей. Одним із найпопулярніших інструментів для розробки ігор є рушій Unity, який надає розробникам різноманітні засоби для інтеграції ШІ в ігровий процес. Впровадження ШІ в ігрові проекти на Unity може значно підвищити якість ігрового досвіду, зменшуючи навантаження на розробників і дозволяючи їм зосередитися на творчих аспектах гри.

Постановка задачі

Провести огляд методів впровадження штучного інтелекту в ігрові проекти на основі рушія Unity з метою визначення найбільш ефективних та продуктивних рішень для покращення ігрового досвіду.

Мета дослідження

Покращення інтеграції штучного інтелекту в ігрові проекти на рушії Unity на основі проведеного огляду методів і технологій.

Результати дослідження

Існує багато різних підходів до впровадження ШІ в ігрові проекти на Unity. Серед найпоширеніших методів - використання поведінкових дерев, машинного навчання, алгоритмів пошуку, нейронних мереж та кінцевих автоматів (Finite State Machines). Кожен з цих методів має свої унікальні переваги та недоліки:

Поведінкові дерева (Behavior Trees) є одним з найбільш популярних підходів, що дозволяє легко моделювати складну поведінку персонажів гри. Вони забезпечують гнучкість і модульність, але можуть бути складними в управлінні для дуже великих і складних проектів [1].

Машинне навчання (Machine Learning) відкриває нові можливості для створення адаптивної та самонавчальної поведінки персонажів. Використання нейронних мереж і методів навчання з підкріпленням може значно покращити реалістичність ігрових агентів, але потребує значних обчислювальних ресурсів та часу на навчання [2].

Алгоритми пошуку (Pathfinding Algorithms) такі як A* є стандартними методами для навігації персонажів в ігровому світі. Вони прості в реалізації та ефективні для багатьох типів ігор, але можуть стати вузьким місцем у дуже великих світах [3].

Нейронні мережі (Neural Networks) дозволяють створювати складні моделі поведінки, які можуть адаптуватися до змін в ігровому середовищі. Вони ефективні для створення більш реалістичних ворогів і NPC, але вимагають значних знань в області машинного навчання [4].

Кінцеві автомати (Finite State Machines, FSM) є простими та ефективними для моделювання поведінки персонажів на основі визначених станів та переходів між ними. Вони легко налаштовуються та зрозумілі для реалізації, але можуть стати громіздкими при великій кількості станів [5].

Порівняльна характеристика методів впровадження ШІ в ігрові проекти на Unity

Метод	Переваги	Недоліки	Використання
Поведінкові дерева	Гнучкість, модульність	Складність в управлінні для великих проектів	Моделювання поведінки персонажів
Машинне навчання	Адаптивність, самонавчання	Високі обчислювальні ресурси, час на навчання	Реалістична поведінка, адаптивні агенти
Алгоритми пошуку	Простота реалізації, ефективність	Можливі вузькі місця у великих світах	Навігація персонажів в ігровому світі
Нейронні мережі	Складні моделі поведінки, адаптивність	Потребують знань в машинному навчанні, обчислювальні ресурси	Створення реалістичних ворогів і NPC
Кінцеві автомати	Простота реалізації, ефективність	Громіздкість при великій кількості станів	Моделювання простих поведінок персонажів

Висновки та перспективи

На основі проведеного огляду можна зробити висновок, що найбільш перспективними методами впровадження ШІ в ігрові проекти на Unity є машинне навчання та нейронні мережі завдяки їхній адаптивності та можливості створення складних моделей поведінки. Комбінація цих методів з поведінковими деревами та кінцевими автоматами може забезпечити високу гнучкість та ефективність. Подальші дослідження та експерименти з комбінацією цих методів можуть привести до значного покращення якості ігрового процесу та задоволення користувачів.

Список використаних джерел:

- 1.How to create a simple behaviour tree in Unity/C# – [Електронний ресурс] – Режим доступу: <https://medium.com/geekculture/how-to-create-a-simple-behaviour-tree-in-unity-c-3964c84c060e> (дата звернення: 20.05.2024).
- 2.Machine Learning in Unity – [Електронний ресурс] – Режим доступу: <https://unity.com/ru/resources/getting-started-with-machine-learning-and-simulation-with-unity> (дата звернення: 20.05.2024).
- 3.Pathfinding Algorithms in Unity – [Електронний ресурс] – Режим доступу: <https://learn.unity.com/project/a-36369ng> (дата звернення: 20.05.2024).
- 4.Neural Networks and AI in Unity – [Електронний ресурс] – Режим доступу: <https://blog.unity.com/engine-platform/real-time-style-transfer-in-unity-using-deep-neural-networks> (дата звернення: 20.05.2024).
- 5.Finite State Machines – [Електронний ресурс] – Режим доступу: <https://learn.unity.com/tutorial/finite-state-machines-1> (дата звернення: 20.05.2024).

Використання рендерингу з багатозональною роздільною здатністю для підвищення продуктивності

*Яковчук Василь Андрійович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
vasyakov4uk@gmail.com
Науковий керівник: Залива Віталій Вікторович,
доктор філософії,
старший викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

З розвитком ігрової індустрії зростають вимоги до якості графіки, що спричиняє збільшенню обчислювального навантаження. Зональний рендеринг — це підхід, який дозволяє оптимізувати використання ресурсів, зосереджуючи високу якість рендерингу на центральній області кадру.

Постановка задачі.

Дослідити метод зонального рендерингу, який використовує різну роздільну здатність у залежності від зони екрану. Головна задача – забезпечити високу продуктивність без втрати якості у центральній зоні. Метод ґрунтується на принципі роботи Variable Rate Shading, описаному Roberts [1].

Мета дослідження.

Використання зонального рендерингу для підвищення продуктивності системи шляхом зниження обчислювального навантаження на графічний процесор. Це досягається за рахунок зменшення кількості пікселів, які потрібно обробити, з одночасним збереженням високої якості у центральній частині кадру [2].

Результати дослідження.

Метод зонального рендерингу дозволив реалізувати поділ екрану на три області: центральну, середню та периферійну (див. рис. 1).

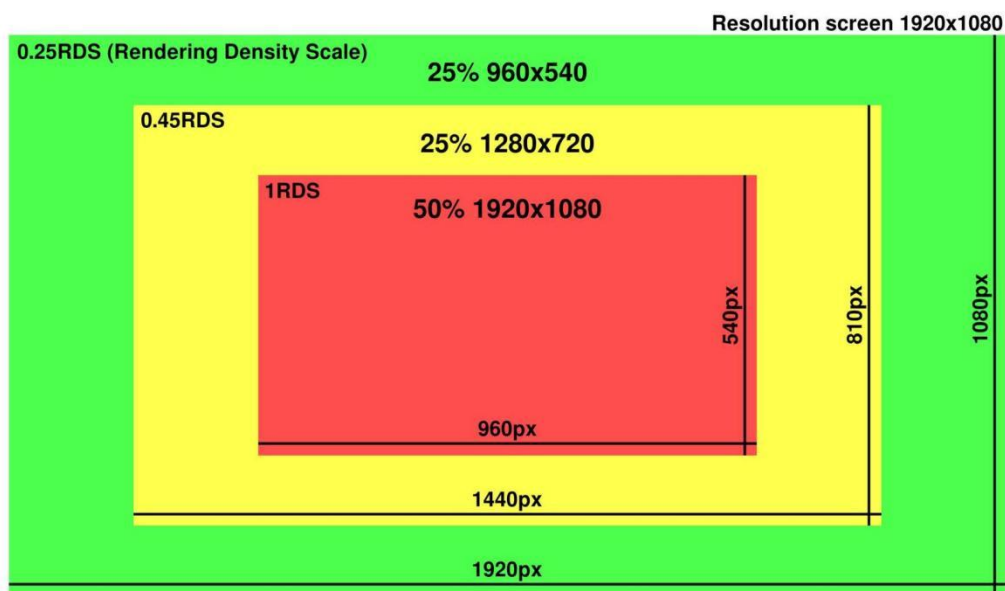


Рис 1. Multi-Region Resolution Rendering (50/25/25)

Центральна частина рендериться у повній якості, тоді як середня та периферійна області мають знижену роздільну здатність. Тестування показало збільшення FPS до 40% у порівнянні зі стандартною роздільною здатністю.

Висновки та перспективи.

Результати цього дослідження свідчать, що метод Multi-Region Resolution Rendering успішно знижує навантаження на GPU та підвищує продуктивність. У перспективі метод може бути розширений до 4-5 зон із модифікацією їхніх форм (див. рис. 2), наприклад, для мобільних ігор. Це дозволить максимально розкрити потенціал методу, враховуючи обмежені графічні потужності пристроїв та часткове перекриття екрану руками.

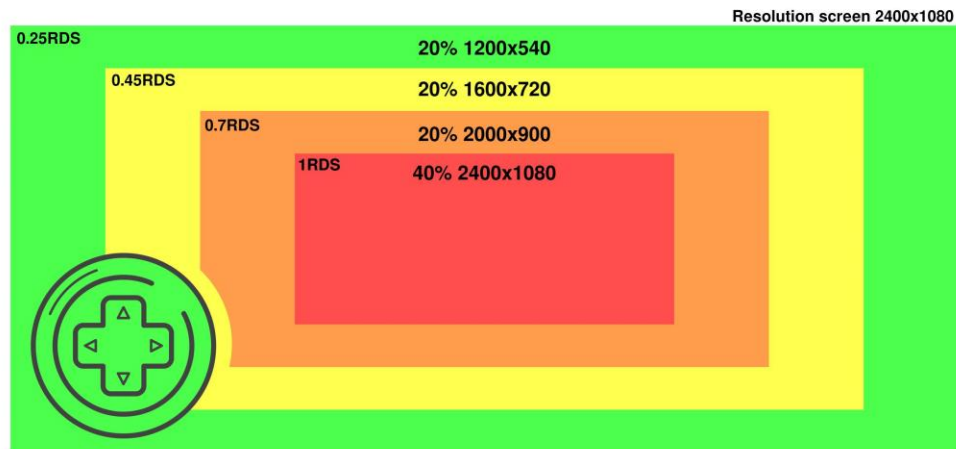


Рис. 2. Multi-Region Resolution Rendering, mobile (40/20/20/20)

Список використаних джерел:

1. Roberts, G. Variable Rate Shading Techniques for Real-Time Applications. SIGGRAPH Asia Technical Briefs, 2019, С. 45.
2. NVIDIA Research. Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination, 2019, С. 28.

Теоретичні основи формування особистісних рис ігрових персонажів

*Ярошевський Олександр Вікторович,
студент групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення
Державного університету інформаційно-комунікаційних технологій
jaroschew@gmail.com
Науковий керівник: Замрій Ірина Вікторівна,
доктор технічних наук, доцент
завідувач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

З розвитком технологій, комп'ютерні ігри стали невід'ємною частиною сучасної культури, а вимоги до їхньої якості та реалістичності зросли. Одна з ключових характеристик, що визначає привабливість гри, — це здатність персонажів динамічно адаптувати свою поведінку до обставин та взаємодії з гравцем. Формування унікальних особистісних рис ігрових персонажів сприяє підвищенню їхньої реалістичності та створенню інтерактивних сценаріїв. У роботі розглядаються теоретичні основи формування особистісних рис персонажів з акцентом на моделі, які можуть бути використані для алгоритмічної реалізації у програмному забезпеченні [1].

Постановка задачі

Сучасні методи моделювання особистісних рис персонажів у комп'ютерних іграх обмежуються передбаченими шаблонами або статичними характеристиками. Завданням даного дослідження є аналіз існуючих психологічних моделей особистості та розробка концепції, що дозволяє застосовувати ці моделі для динамічного формування особистісних рис ігрових персонажів, які можуть впливати на їхні рішення в ігровому середовищі [2].

Мета дослідження

Метою даного дослідження є розробка теоретичної основи для створення ігрових персонажів із реалістичними особистісними рисами, що сприятиме підвищенню інтерактивності та адаптивності віртуальних середовищ. Це передбачає визначення та вибір відповідних моделей особистості, а також обґрунтування їхньої інтеграції у процес розробки ігрових персонажів.

Результати дослідження

Результати дослідження охоплюють огляд і адаптацію основних моделей особистості для застосування в ігрових персонажах:

Модель "Великої п'ятірки" (Big Five).

Ця модель включає п'ять основних рис: екстраверсію, доброзичливість, сумлінність, емоційну стабільність та відкритість до нового досвіду. Модель широко використовується, оскільки її параметри дозволяють створювати персонажів із варіативними, але добре зрозумілими рисами, які легко змінювати залежно від ігрових ситуацій [3].

Модель HEXACO.

Це розширена версія Big Five, яка додає шосту рису — чесність-смирнення, що дозволяє створювати персонажів із різними моральними орієнтирами та поведінковими сценаріями. Модель HEXACO підходить для ігор з акцентом на моральний вибір і дозволяє більш комплексно відобразити поведінку персонажів у складних емоційних або етичних ситуаціях [4].

Модель Майєрс-Бріггс (МВТІ).

Базується на поєднанні 16 особистісних типів, розподілених за шкалами екстраверсія/інтроверсія, сенсорика/інтуїція, мислення/відчуття та прийняття рішень. Вона забезпечує глибокий рівень деталізації, що підходить для персонажів із багат шаровою поведінкою, однак через свою складність ця модель не завжди зручна для програмної реалізації.

Теорія особистісних потреб Маслоу або Альдерфера.

Описує поведінку персонажів через ієрархію потреб, починаючи від базових (фізіологічних) до вищих (самореалізація). Ця модель добре підходить для ігор, де персонажі реагують на задоволення або незадоволення різних рівнів потреб, але більше підходить для створення сценаріїв, ніж формування характеру [5].

"Темна тріада" (Dark Triad).

Охоплює три риси: нарцисизм, макіавеллізм та психопатію. Вона допомагає створювати персонажів із негативними або неоднозначними рисами, проте її застосування обмежене іграми, що потребують антагоністичних чи складних у взаємодії персонажів [6].

Висновки та перспективи

Інтеграція кількох теоретичних моделей особистості, зокрема Big Five, HEXACO та Dark Triad створює надійну основу для алгоритмічного формування особистісних рис ігрових персонажів. Big Five забезпечує зрозумілу і гнучку базу для налаштування поведінки NPC, тоді як HEXACO додає можливість враховувати моральні та етичні аспекти у діях персонажів. Можна також включити концепцію Dark Triad. Хоча вона і має негативний характер, проте є важливою для моделювання поведінки персонажів у іграх, особливо коли йдеться про створення складних, емоційно багатих персонажів.

Пропонується використати комбінацію цих моделей для динамічного моделювання поведінки персонажів, де їхні особистісні риси змінюються відповідно до ігрових ситуацій. Такий підхід забезпечить високий рівень реалістичності, підвищуючи інтерактивність і варіативність взаємодії з гравцем та іншими NPC, що є новим кроком у розробці адаптивних ігрових сценаріїв.

Список використаних джерел:

1. Player behavioural modelling for video games - sciencedirect.com [Електронний ресурс] – Режим доступу : <https://www.sciencedirect.com/science/article/abs/pii/S1875952111000486> (дата звернення: 15.08.2024).
2. Player behavior and personality modeling for interactive storytelling in games - sciencedirect.com [Електронний ресурс] – Режим доступу : <https://www.sciencedirect.com/science/article/abs/pii/S1875952118300120?via%3Dihub> (дата звернення: 15.08.2024).
3. Evaluating the Big Five as an organizing framework for commonly used psychological trait scales - psycnet.apa.org [Електронний ресурс] – Режим доступу : <https://psycnet.apa.org/buy/2022-21018-001> (дата звернення: 16.08.2024).
4. Big Five and HEXACO Personality Traits, Proenvironmental Attitudes, and Behaviors: A Meta-Analysis - journals.sagepub.com [Електронний ресурс] – Режим доступу : <https://journals.sagepub.com/doi/full/10.1177/1745691620903019> (дата звернення: 16.08.2024).
5. Using the Myers-Briggs Type Indicator (MBTI) for Modeling Multiagent Systems - seer.ufrgs.br [Електронний ресурс] – Режим доступу : https://seer.ufrgs.br/rita/article/view/RITA_Vol29_Nr1_42 (дата звернення: 17.08.2024).
6. A primer on the Dark Triad traits - tandfonline.com [Електронний ресурс] – Режим доступу : <https://www.tandfonline.com/doi/full/10.1111/ajpy.12198#abstract> (дата звернення: 17.08.2024).

СЕКЦІЯ 6. ОПТИМІЗАЦІЯ БІЗНЕС ПРОЦЕСІВ

Оптимізація керування даними в односторінкових веб-додатках через адаптивне кешування

*Власенко Іван Юрійович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
ivanvlaskenko206@gmail.com*

*Науковий керівник: Залива Віталій Вікторович,
доктор філософії (PhD),
ст. викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Односторінкові веб-додатки (SPA) набувають все більшого поширення в сучасних цифрових екосистемах завдяки інтерактивності та швидкодії. Водночас, із зростанням обсягів даних та складності функціоналу цих додатків постають питання ефективності їх роботи, зокрема у сфері обробки та передачі даних. Недостатня оптимізація процесів управління даними може призводити до перевантаження клієнтської частини, зниження швидкодії та значних затримок у користувацькому досвіді. Вирішення цих проблем є актуальним завданням у контексті забезпечення масштабованості та продуктивності SPA [1-3].

Постановка задачі.

Розробити та впровадити метод підвищення ефективності управління даними в SPA шляхом мінімізації часу завантаження та обсягу переданої інформації, а також зниження кількості звернень до серверної частини. Завданням дослідження є створення алгоритму, який забезпечить адаптивне кешування даних у реальному часі, що дозволить оптимізувати обробку та зберігання даних на клієнтській частині додатків.

Мета дослідження.

Розробити та впровадити алгоритм адаптивного кешування даних для односторінкових веб-додатків, який враховує динаміку змін даних та залежності між компонентами, з метою зменшення часу завантаження, оптимізації мережевих витрат і забезпечення стабільної роботи додатків навіть в умовах обмеженої пропускної здатності мережі.

Результати дослідження.

У ході дослідження було розроблено алгоритм адаптивного кешування, який включає такі ключові аспекти:

1. Аналіз графів залежностей: Використання графів залежностей для визначення змінених компонентів даних та зменшення обсягів переданої інформації між клієнтом і сервером.
2. Розподіл даних за типами: Класифікація даних на статичні, динамічні та проміжні, що дозволяє застосовувати різні стратегії оновлення залежно від частоти змін та важливості даних.
3. Автоматизація налаштувань кешу: Впровадження механізмів динамічного коригування часу життя кешу (TTL) та політик оновлення на основі поведінки користувачів і аналізу запитів. [4]

Впровадження цього механізму дозволило:

- Зменшення кількості запитів до сервера на 40%, що знизило навантаження на серверну інфраструктуру.
- Скорочення середнього часу завантаження сторінки на 35%, що покращило продуктивність додатка.
- Підвищення стабільності роботи додатка при високих навантаженнях завдяки ефективному управлінню кешуванням.
- Зменшення обсягів переданої інформації на 25% за рахунок оптимізації вибіркового оновлення даних.

Висновки та перспективи.

Результати дослідження підтвердили ефективність запропонованого підходу до управління даними в SPA. Розроблений алгоритм адаптивного кешування сприяє покращенню продуктивності веб-додатків, зменшенню навантаження на мережу та сервери, що є важливим для забезпечення стабільності бізнес-процесів, які базуються на SPA. Подальші дослідження спрямовані на інтеграцію алгоритму з методами машинного навчання для прогнозування змін у даних та автоматизації оптимізації кешування з урахуванням поведінки користувачів.

Список використаних джерел:

1. Mastering JavaScript Single Page Application Development. Packt Publishing - ebooks Account, 2016. 452 p.
2. Shivakumar S. K. Modern Web Performance Optimization: Methods, Tools, and Patterns to Speed up Digital Platforms. Apress L. P., 2020.
3. Shivakumar S. K. Modern Web Performance Optimization: Methods, Tools, and Patterns to Speed up Digital Platforms. Apress L. P., 2020.
4. What Is Time to Live (TTL)? - RocketCDN. RocketCDN. URL: <https://rocketcdn.me/what-is-time-to-live-ttl/> (дата звернення: 23.11.2024).

Огляд розробки алгоритмів для автоматичного виявлення та оцінки ризиків у проектному управлінні

*Златоус Олексій Дмитрович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
generalgrevus@gmail.com
Науковий керівник: Задонцев Юрій Вікторович,
кандидат технічних наук, доцент,
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Ризики – це невід’ємна частина робочих систем та процесів. Вони можуть виникати на будь-якому етапі, починаючи від планування та закінчуючи реалізацією проекту. Управління ризиками включає ідентифікацію, оцінку та вжиття заходів щодо зниження їх негативного впливу.

Сучасне управління проектами стикається з необхідністю аналізу та мінімізації ризиків. Управління ризиками є ключовим елементом, який сприяє досягненню цілей проекту у встановлені терміни та в рамках бюджету. Автоматизація процесу виявлення та оцінки ризиків з використанням алгоритмів машинного навчання та аналізу даних стає актуальним завданням, що сприяє підвищенню ефективності управління проектами, а також мінімізацією можливої шкоди. Автоматичні системи управління ризиками можуть значно скоротити час на аналіз та підвищити точність прогнозів, що є критично важливим для успішного завершення проектів.

Постановка задачі

Необхідність автоматизації процесів виявлення та оцінки ризиків у проектному управлінні продиктована прагненням підвищити точність та швидкість прийняття рішень, а також знизити вплив суб’єктивних факторів. Проектне управління включає безліч аспектів, таких як планування, виконання завдань, контроль за ресурсами і бюджетом. Однак одним із ключових завдань є управління ризиками, які можуть виникнути на будь-якому етапі проекту.

Метою даної є огляд застосування алгоритмів для автоматичного виявлення та оцінки ризиків в проектному управлінні, спрямованих на підвищення точності та оперативності прийняття управлінських рішень.

Мета дослідження

Вивчити теоретичні та практичні аспекти розробки та впровадження алгоритмів для автоматичного виявлення та оцінки ризиків у проектному управлінні. Дослідити існуючі методи та алгоритми, що застосовуються для аналізу ризиків, та запропонувати нові підходи, що забезпечують більш високу точність та оперативність.

Результати дослідження

У ході дослідження було розроблено та протестовано алгоритми, засновані на передових методах машинного навчання, таких як регресійні моделі, дерева рішень та нейронні мережі, з метою автоматизації процесів виявлення та оцінки ризиків у проектному управлінні.

Основні завдання дослідження включали.

1. Збір та попередня обробка даних проекту:
 - Алгоритми були налаштовані на автоматичне збирання та обробку даних проектів, включаючи історичні дані та поточні показники.
2. Ідентифікація потенційних ризиків:
 - На основі аналізу зібраних даних виявлено потенційні ризики, які можуть виникнути на різних етапах проекту.
3. Оцінка ймовірності та впливу ризиків:
 - Розроблені алгоритми дозволили не лише визначити ймовірність настання кожного ризику, а й оцінити потенційний вплив на проектні цілі та терміни.
4. Ранжування ризиків за пріоритетністю:
 - Завдяки використанню методів машинного навчання було здійснено автоматичну класифікацію та ранжування ризиків, що дозволило виявити найбільш критичні загрози для проекту.

Оцінка ефективності алгоритмів:

Алгоритми були протестовані на реальних даних кількох проектів, що дозволило отримати такі результати:

- Скорочення часу на аналіз ризиків на 30%:
 - Автоматизація процесів виявлення та оцінки ризиків значно знизилася час, що витрачається на аналіз, завдяки оперативному та точному прогнозуванню потенційних загроз.
- Поліпшення точності прогнозів на 25%:
 - Використання передових алгоритмів машинного навчання значно підвищило точність оцінки ризиків, що сприяє більш обґрунтованому прийняттю управлінських рішень та мінімізації негативних впливів на проектні показники.

Ці результати підтверджують ефективність та потенціал використання алгоритмів машинного навчання у сучасному проектному управлінні для підвищення оперативності та точності управлінських рішень.

Наведу кілька прикладів галузей

1. Управління будівельними проектами

- **Система:** Procore – популярна платформа для управління будівельними проектами.
- **Методи:** Впровадження регресійних моделей та дерев рішень для оцінки ризиків затримок.
- **Результати:** Прогнозування ризиків затримок було покращено на 25% завдяки використанню історичних даних про погодні умови, постачання матеріалів та попередні проекти. Середній час відгуку виявлення ризиків скоротилося з 7 днів до 3 днів.

2. IT-проекти

- **Система:** Jira - інструмент, що широко використовується для управління проектами у сфері розробки програмного забезпечення.

- **Методи:** Застосування нейронних мереж для аналізу метрик розробки (кількість багів, продуктивність команди, зміна вимог).
- **Результати:** Прогнозування ризиків порушення термінів було покращено на 30%. У результаті кількість проектів, завершених у строк, збільшилась на 15%, а перевитрата бюджету зменшилась на 10%.

3. Фінансові проекти

- **Система:** SAS Risk Management – платформа для управління фінансовими ризиками.
- **Методи:** Використання моделей машинного навчання для оцінки кредитних ризиків на основі даних про платоспроможність клієнтів та економічні показники.
- **Результати:** Точність оцінки кредитних ризиків було підвищено на 20%. Це дозволило скоротити кількість невиконаних кредитних зобов'язань на 12% та збільшити прибуток на 8%.

Приклади застосування методів та технологій

- **Регресійні моделі:** Використовувалися в Rcosore для прогнозування затримок. Лінійна регресія дозволила врахувати вплив різних факторів (погодні умови, постачання матеріалів) на ймовірність затримок.
- **Дерева рішень та випадкові ліси:** Застосовувалися в Jira для класифікації ризиків. Ці методи дозволили врахувати численні взаємозв'язки між факторами ризику та їх впливом геть терміни і бюджет проектів.
- **Нейронні мережі:** Впроваджувалися в систему SAS Risk Management для аналізу складних патернів у даних про платоспроможність клієнтів. Глибокі нейронні мережі (DNN) та рекурентні нейронні мережі (RNN) використовувалися для прогнозування часових рядів та оцінки тимчасових ризиків.

- **Обробка природної мови (NLP):** Застосовувалась в Jira для аналізу текстових даних зі звітів та коментарів. Методи NLP дозволили виявляти потенційні ризики на основі аналізу тональності та змісту текстів.

Критичні зони та виклики

Незважаючи на значні досягнення, були виявлені й критичні зони, які потребують уваги:

- **Якість даних:** Недостатньо якісні або неповні дані можуть суттєво вплинути на точність моделей. У проектах часто виникають проблеми з даними через їхню різноманітність і неузгодженість.
- **Складність моделей:** Складні моделі машинного навчання можуть бути складними в інтерпретації та поясненні. Це ускладнює прийняття рішень на їх основі та вимагає додаткових зусиль щодо розробки методів візуалізації та зрозумілості моделей.
- **Інтеграція в існуючі системи:** Впровадження нових алгоритмів потребує значних зусиль щодо інтеграції з існуючими системами управління проектами. Це може вимагати додаткових витрат та часу на адаптацію.

Висновки та перспективи.

Автоматизація процесу управління ризиками з використанням алгоритмів машинного навчання та аналізу даних відкриває нові можливості для підвищення ефективності управління проектами. Запропоновані алгоритми дозволяють оперативно виявляти та оцінювати ризики, що сприяє прийняттю своєчасних та обґрунтованих управлінських рішень.

Перспективи подальших досліджень включають:

- Удосконалення алгоритмів з урахуванням специфіки різних галузей та типів проєктів.
- Розробку інтеграційних рішень для впровадження алгоритмів у існуючі системи управління проектами.

- Дослідження можливостей використання алгоритмів для прогнозування не лише ризиків, а й можливих можливостей та сприятливих сценаріїв розвитку проекту.

Висновок

Розробка алгоритмів для автоматичного виявлення та оцінки ризиків у проектному управлінні є важливим кроком на шляху до покращення управління проектами. Автоматизація цього процесу сприяє більш ефективному використанню ресурсів та підвищенню ймовірності успішного завершення проектів у встановлені терміни та в рамках бюджету.

Список використаних джерел

1. Bissonette, M. M. Project Risk Management: A Practical Implementation Approach [Електронний ресурс] / Michael M. Bissonette // Сторінки: 23-45. – Режим доступу: <https://www.perlego.com/book/1354924/project-risk-management-a-practical-implementation-approach-pdf>
2. Ng, A. Machine Learning Yearning [Електронний ресурс] / Andrew Ng // Сторінки: 56-78. – Режим доступу: https://nessie.ilab.sztaki.hu/~kornai/2020/AdvancedMachineLearning/Ng_MachineLearningYearning.pdf
3. Provost, F., Fawcett, T. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking [Електронний ресурс] / Foster Provost, Tom Fawcett // Сторінки: 102-126. – Режим доступу: https://www.researchgate.net/publication/256438799_Data_Science_for_Business
4. Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow [Електронний ресурс] / Aurélien Géron // Сторінки: 145-167. - Режим доступу: <https://ebin.pub/scikit-learn-keras-tensorflow-2nbsped-9785907203334.html>

Розробка чат-боту для автоматизації клієнтської підтримки у сфері малого та середнього бізнесу

*Колодюк Андрій Васильович,
аспірант 3 курсу,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
a.kolodiuk@duikt.edu.ua
Науковий керівник: Жебка Вікторія Вікторівна,
д.т.н., професор
Державного університету інформаційно-комунікаційних технологій,*

В умовах зростаючої цифровізації бізнес-процесів чат-боти стають невід'ємною частиною сучасних клієнтських сервісів. Вони дозволяють автоматизувати значну частину рутинної роботи, забезпечуючи при цьому цілодобову підтримку. Дослідження спрямоване на створення та тестування прототипу чат-бота, що інтегрується з реальними бізнес-процесами.

Постановка задачі.

Задачею дослідження є розробка чат-боту, який здатен автоматизувати відповіді на часті запити клієнтів.

Мета дослідження.

Метою є створення практичного рішення у вигляді прототипу чат-бота, яке зменшить навантаження на персонал, покращить якість обслуговування клієнтів та оптимізує витрати компанії.

Результати дослідження.

Згідно з сучасними дослідженнями найпопулярніших платформ для надання клієнтського сервісу, таких як Zendesk, Tidio, більшість запитів до служби підтримки в бізнесі є рутинними та можуть бути автоматизовані за допомогою чат-ботів. Рутинні запити становлять до 80% всіх запитів до служби підтримки, серед яких поширеними є запити про доставку, статус замовлень, повернення товарів. Дані ресурсів Missive, Dashley свідчать про те, що чат-боти можуть ефективно автоматизувати до 69% таких рутинних запитів.

Вибрано платформу Dialogflow для обробки природної мови (NLP) та інтеграції з вебсайтом компанії. Для збереження клієнтських даних передбачено інтеграцію з CRM-системою.

Створено базові сценарії відповідей для типових запитів клієнтів, включаючи інформацію про доступність товару, інструкції щодо повернення, відслідковування замовлення, доступ до персоналізованих даних клієнта через CRM.

В якості додаткової інтеграції з бізнес-системами реалізовано перенаправлення складних запитів до операторів.

Висновки та перспективи.

Очікується, що даний продукт збільшить швидкість обслуговування клієнтів, підвищить рівень задоволеності споживачів, заощадить фінансові та людські ресурси компанії, адже зменшаться витрати на навчання нових операторів та утримання великого штату персоналу.

Крім того, прототип має модульну структуру, що дозволить адаптувати його для різних сфер діяльності в галузі малого та середнього бізнесу. Однак важливо не лише впроваджувати технологію, а й адаптувати її до чітко визначених потреб конкретного бізнес-кейсу.

Отже, розробка чат-ботів є ефективним рішенням для автоматизації клієнтської підтримки. Це дозволяє зменшити навантаження на персонал, підвищити якість обслуговування клієнтів і зміцнити конкурентні позиції бізнесу.

Список використаних джерел:

1. М.О. Чупріна, А.А. Орозонова, "Використання технологій чат-ботів в умовах цифрової трансформації бізнесу", Матеріали конференції КПІ 2021: II Міжнародна науково-практична конференція «Бізнес, інновації, менеджмент: проблеми та перспективи» / Секція 2. Менеджмент міжнародного бізнесу в умовах глобалізації. – [Електронний ресурс] – Режим доступу: <https://confmanagement-proc.kpi.ua/article/view/230945>.

2. О.А. Біловодська, К.О. Лагута, "Системне дослідження використання чат-боту в комунікації з клієнтами" // Формування ринкових відносин в Україні. – 2020. – № 5 (228). – С. 62-68. – [Електронний ресурс] – Режим доступу: <https://er.knutd.edu.ua/handle/123456789/15953>.
3. О. Небилиця, "Чат-боти як ефективний стратегічний та інноваційний напрям розвитку бізнесу", Збірник наукових праць ХНЕУ ім.С.Кузнеця, № 2(85), 2021. – [Електронний ресурс] – Режим доступу: https://scholar.google.com/citations?view_op=view_citation&hl=ru&user=7Vtq6vEAAA AJ&citation_for_view=7Vtq6vEAAA AJ:YsMSGLbcyi4C.
4. "Як чат-боти зі штучним інтелектом змінюють клієнтську підтримку", RankTracker Blog, 2023. – [Електронний ресурс] – Режим доступу: <https://www.ranktracker.com/uk/blog/how-ai-powered-chatbots-are-transforming-customer-support/>.

Підвищення ефективності розробки програмного забезпечення шляхом оптимізації процесів обробки ресурсів

*Фетько Юрій Іванович,
студент групи ПДМ-63,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
fetko.yura@gmail.com
Науковий керівник: Залива Віталій Вікторович,
доктор філософії (PhD),
ст.викладач кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

У сучасному світі розробки програмного забезпечення оптимізація бізнес-процесів відіграє ключову роль у забезпеченні конкурентоспроможності ІТ-компаній. Одним із критичних аспектів є ефективність процесу розробки, зокрема оптимізація обробки та збірки ресурсів проекту. За статистикою, розробники витрачають до 30% робочого часу на очікування компіляції та обробки файлів проекту [2-4].

Постановка задачі.

Дослідження та розробка методів оптимізації бізнес-процесів розробки програмного забезпечення шляхом впровадження інтелектуальної системи обробки ресурсів проекту [4].

Мета дослідження.

Підвищення ефективності бізнес-процесів розробки програмного забезпечення за рахунок скорочення часу на обробку ресурсів проекту та оптимізації використання обчислювальних ресурсів [1].

Результати дослідження.

Проведено аналіз існуючих бізнес-процесів розробки та виявлено критичні точки, що впливають на ефективність роботи команди:

5. Тривалий час очікування обробки ресурсів
6. Неefективне використання обчислювальних потужностей
7. Надмірне споживання системних ресурсів
8. Затримки в процесі розробки

Запропоновано комплексне рішення, що включає:

- Впровадження системи інкрементної обробки файлів
- Оптимізацію процесів кешування
- Автоматизацію управління залежностями
- Паралельну обробку ресурсів

За результатами впровадження розробленого рішення досягнуто:

- Скорочення часу обробки ресурсів на 40%
- Зменшення витрат на обчислювальні ресурси на 15%
- Підвищення продуктивності команди розробників на 10%

Висновки та перспективи.

Запропоноване рішення демонструє значний потенціал для оптимізації бізнес-процесів розробки програмного забезпечення. Економічний ефект від впровадження системи проявляється у скороченні витрат на обчислювальні ресурси та підвищенні продуктивності команди розробників.

Подальші дослідження можуть бути спрямовані на розширення функціоналу системи та інтеграцію з іншими інструментами оптимізації бізнес-процесів.

Список використаних джерел:

- 1.Чумаченко І. В. Управління проектами розробки програмного забезпечення: підручник / І. В. Чумаченко, Н. В. Доценко. – Харків: ХНУМГ ім. О. М. Бекетова, 2023. – 295 с.
- 2.State of DevOps Report 2023 [Електронний ресурс] // Puppet Labs. – 2023. – Режим доступу: <https://puppet.com/resources/report/state-of-devops-report>
- 3.Лавріщева К. М. Програмна інженерія: підручник для студентів вищих навчальних закладів / К. М. Лавріщева. – К.: Академперіодика, 2022. – 456 с.
- 4.Web Performance Working Group [Електронний ресурс] // W3C. – 2023. – Режим доступу: <https://www.w3.org/webperf/>

Гріді-алгоритми на класах (ψ, β) -диференційовних періодичних функцій

*Замрій Ірина Вікторівна,
доктор технічних наук, доцент,
завідувач кафедри Інженерії програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
irinafraktal@gmail.com*

*Шкапа Вікторія Вікторівна,
кандидат фізико-математичних наук, доцент,
доцент кафедри Вищої математики, математичного моделювання та фізики,
Державного університету інформаційно-комунікаційних технологій
vshkapa@ukr.net*

Розглядається наближення періодичних функцій із класів $L_{\beta,p}^{\psi}$ у просторі L_q .
Класи $L_{\beta,p}^{\psi}$ було введено О. І. Степанцем (див., наприклад, [1]). Зазначимо, що у випадку $\psi(|k|) = |k|^{-r}, r > 0, k \in \mathbb{Z} \setminus \{0\}$ вони співпадають з класами Вейля-Надя $W_{\beta,p}^r$ (див., наприклад, [1]).

Через B позначимо множину додатних і незростаючих функцій $\psi(t)$,
 $t \in \mathbb{N}$, для кожної з яких існує стала $C > 0$ така, що $\frac{\psi(t)}{\psi(2t)} \leq C, t \in \mathbb{N}$.

L_q – простір вимірних 2π -періодичних функцій f зі стандартною нормою.

Нехай $\{\hat{f}(k(l))\}_{l=1}^{\infty}$ — коефіцієнти Фур'є $\{\hat{f}(k)\}_{k \in \mathbb{Z}}$ функції $f \in L_1$,
впорядковані у порядку незростання їх модулів, тобто

$$|\hat{f}(k(1))| \geq |\hat{f}(k(2))| \geq \dots$$

Позначимо для $f \in L_q$:

$$G_m(f, x) := \sum_{l=1}^m \hat{f}(k(l)) e^{ik(l)x}$$

і розглянемо величину

$$\sup_{f \in F} \|f(\cdot) - G_m(f, \cdot)\|_q.$$

Для функціонального класу $F \subset L_q$ покладемо

$$G_m(F)_q := \sup_{f \in F} \| f(\cdot) - G_m(f, \cdot) \|_q. \quad (1)$$

Величину (1) називають гріди-алгоритмом (від англ. *greedy algorithm*). З історією дослідження даної величини для деяких важливих функціональних класів можна ознайомитися в роботі [2].

Мають місце такі твердження:

Теорема 1. Нехай $1 \leq q \leq 2 \leq p < \infty$, $\psi \in B$, $\beta \in \mathbb{R}$. Тоді справедлива порядкова оцінка

$$G_m(L_{\beta,p}^\psi)_q \asymp \psi(m).$$

Теорема 2. Нехай $1 < p < q \leq 2$, $\psi \in B$, $\beta \in \mathbb{R}$ і, крім того, існує $\varepsilon > 0$ таке, що послідовність $\psi(t)t^{\frac{1}{p}-\frac{1}{q}+\varepsilon}$, $t \in \mathbb{N}$, не зростає. Тоді справедлива порядкова оцінка

$$G_m(L_{\beta,p}^\psi)_q \asymp \psi(m)m^{\frac{1}{p}-\frac{1}{2}}.$$

Теорема 3. Нехай $1 < p \leq 2 \leq q < \infty$, $\psi \in B$, $\beta \in \mathbb{R}$ і, крім того, існує $\varepsilon > 0$ таке, що послідовність $\psi(t)t^{\frac{1}{p}+\varepsilon}$, $t \in \mathbb{N}$, не зростає. Тоді справедливе співвідношення

$$G_m(L_{\beta,p}^\psi)_q \asymp \psi(m)m^{\frac{1}{p}-\frac{1}{q}}.$$

Теорема 4. Нехай $2 \leq p \leq q < \infty$, $\psi \in B$, $\beta \in \mathbb{R}$ і, крім того, існує $\varepsilon > 0$ таке, що послідовність $\psi(t)t^{\frac{1}{2}+\varepsilon}$, $t \in \mathbb{N}$, не зростає. Тоді справедливе співвідношення

$$G_m(L_{\beta,p}^\psi)_q \asymp \psi(m)m^{\frac{1}{2}-\frac{1}{q}}.$$

Теорема 5. Нехай $2 \leq q < \infty$, $\psi \in B$, $\beta \in \mathbb{R}$, виконується одна з умов

$$\Delta^2\left(\frac{1}{\psi(k)}\right) \geq 0, k \in N, \text{ або } \Delta^2\left(\frac{1}{\psi(k)}\right) \leq 0, k \in N,$$

$$\text{де } \Delta^2\left(\frac{1}{\psi(k)}\right) = \frac{1}{\psi(k)} - \frac{2}{\psi(k+1)} + \frac{1}{\psi(k+2)},$$

і, крім того, існує $\varepsilon > 0$ таке, що послідовність $\psi(t)t^{1+\varepsilon}$, $t \in \mathbb{N}$, не зростає. Тоді справедлива порядкова оцінка

$$G_m(L_{\beta,1}^\psi)_q \approx \psi(m)t^{1-\frac{1}{q}}.$$

Зауваження. Поклавши у теоремах 1 – 5 $\psi(|k|) = |k|^{-r}$, отримаємо відповідні результати для величин $G_m(W_{\beta,p}^r)_q$, які раніше отримано В. М. Темляковим у роботі [2].

Список використаних джерел:

1. Stepanets A.I. Classification and Approximation of Periodic Functions. – London, 1995.
2. Temlyakov V. N. Greedy approximation. – Cambridge: Cambridge University Press. – 2011. – 418 p.

Концептуальна та математична модель адаптивного алгоритму маршрутизації для С2С логістики із застосуванням глибокого навчання

*Коваленко Данило Сергійович
аспірант 4 курсу, групи АКІ-123*

*Державного університету інформаційно-комунікаційних технологій
kds.contactbox@gmail.com*

*Науковий керівник: Лащевська Наталія Олександрівна,
кандидат технічних наук, доцент,*

*завідувач кафедри Комп'ютерної інженерії
Державного університету інформаційно-комунікаційних технологій*

Постановка задачі.

Розвиток електронної комерції, особливо в сегменті С2С логістики, привів до необхідності створення нових підходів до маршрутизації доставки. Цей тип логістики характеризується прямою взаємодією між відправниками і отримувачами, що створює додаткові виклики, такі як висока динамічність попиту, неоднорідність умов середовища та обмеженість ресурсів для ефективного управління [1].

Традиційні підходи до маршрутизації базуються на статичних або напівстатичних алгоритмах, які не враховують реальні зміни у попиті, заторах чи погодних умовах [2]. Це значно обмежує їхню ефективність у сучасних динамічних умовах. Технології глибокого навчання (DL) та підкріплювального навчання (RL) пропонують інструменти для створення систем, що здатні адаптуватися до змін середовища в режимі реального часу [3]. Проте інтеграція цих методів у маршрутизацію для С2С логістики залишається недостатньо дослідженою [4].

Мета дослідження.

Метою даного дослідження є розробка концептуальної та математичної моделей адаптивного алгоритму маршрутизації, який:

1. Інтегрує методи глибокого навчання для точного прогнозування попиту на основі великих даних [1].
2. Використовує підкріплювальне навчання для динамічної адаптації маршрутів, враховуючи змінність середовища [3].
3. Забезпечує масштабованість для роботи у великих містах із високою щільністю доставки [4].

Кінцевою метою є створення основи для майбутньої розробки методики впровадження, яка включатиме практичні аспекти роботи алгоритму в реальних умовах.

Результати дослідження.

Розроблено концептуальну модель адаптивного алгоритму маршрутизації, яка поєднує прогнозування попиту (на основі LSTM і CNN) та підкріплювальне навчання для маршрутизації. Модель описує процес інтеграції прогнозованих даних із механізмами RL, що дозволяє агентам приймати оптимальні рішення в умовах змінного середовища [2, 3].

Формалізовано задачу маршрутизації у вигляді Марковських процесів прийняття рішень (MDP). Ця модель враховує динамічні стани системи, можливі дії

агентів та функцію винагороди, оптимізовану для зменшення часу доставки, витрат і підвищення ефективності використання ресурсів [4].

Висновки та перспективи.

Розроблені концептуальна та математична моделі забезпечують теоретичну базу для створення адаптивних алгоритмів маршрутизації, які відповідають потребам сучасної С2С логістики. Їх впровадження дозволить покращити точність, швидкість прийняття рішень та ефективність використання ресурсів [1, 3].

У перспективі планується:

1. Розробка методики практичної реалізації алгоритмів у реальних логістичних середовищах.
2. Розширення функціоналу для роботи з автономними транспортними засобами.
3. Проведення масштабованих симуляцій із застосуванням реальних даних великих міст [4].

Список використаних джерел

1. Reinforcement learning for logistics and supply chain management Methodologies, state of the art, and future opportunities / Yan Y., Chow AHF., Ho CP., Kuo YH., Wu Q., Ying C. // *Transportation Research Part E: Logistics and Transportation Review* – 2022. – Режим доступу до ресурсу: <https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=3935816>
2. Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment / Ye Y., Zhang X., Sun J. // *Transportation Research Part C: Emerging Technologies* – 2019. – Режим доступу до ресурсу: <https://arxiv.org/pdf/1804.06264>
3. Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications / Ibrahim S., Mostafa M., Jnadi A., Salloum H., Osinenko P. // *IEEE Access* – 2024. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/iel8/6287639/6514899/10763475.pdf>
4. Convex Q-Learning in a Stochastic Environment / Lu F., Mayen S. // *IEEE Conference on Decision and Control* – 2023. – Режим доступу до ресурсу: <https://doi.org/10.48550/arXiv.2309.05105>

Розробка методики та засобів підтримки прийняття рішень для управління проектами на основі аналізу ризиків

*Златоус Олексій Дмитрович,
студент групи ПДМ-61,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
generalgrevus@gmail.com
Науковий керівник: Задонцев Юрій Вікторович,
кандидат технічних наук,
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

Сучасне управління проектами у сфері ІТ стикається з багатьма ризиками, пов'язаними із строками, бюджетом, якістю виконання та іншими факторами. Ефективний аналіз та управління ризиками є критичним для успішного завершення проектів. Комбінація методів Монте-Карло та Байєсівської статистики може запропонувати більш точний підхід до оцінки та прогнозування ризиків у реальному часі.

Постановка задачі.

Основною задачею дослідження є розробка інтегрованої методики аналізу ризиків, яка поєднує методи Монте-Карло та Байєсівське оновлення. Така методика повинна враховувати специфіку ІТ-проектів, забезпечуючи ефективне прийняття рішень.

Мета дослідження.

Метою дослідження є розробка гнучкої архітектури для комбінованого підходу аналізу ризиків, що дозволяє адаптувати моделі до змінних умов проекту і підвищити точність прогнозування ризиків.

Результати дослідження.

У процесі роботи було розроблено архітектуру моделі, яка включає кілька етапів: збір даних, моделювання за методом Монте-Карло, коригування даних за допомогою Байєсівської статистики та візуалізацію результатів. Для реалізації

використовувалися сучасні технології, такі як Python і бібліотеки для статистичного аналізу та моделювання (наприклад, NumPy, PyMC3).

Переваги комбінованого підходу були підтвержені на основі моделювання ризиків у гіпотетичному IT-проекті. Методика дозволила адаптивно оновлювати оцінки ризиків при зміні вхідних даних, що значно підвищило точність прогнозів.

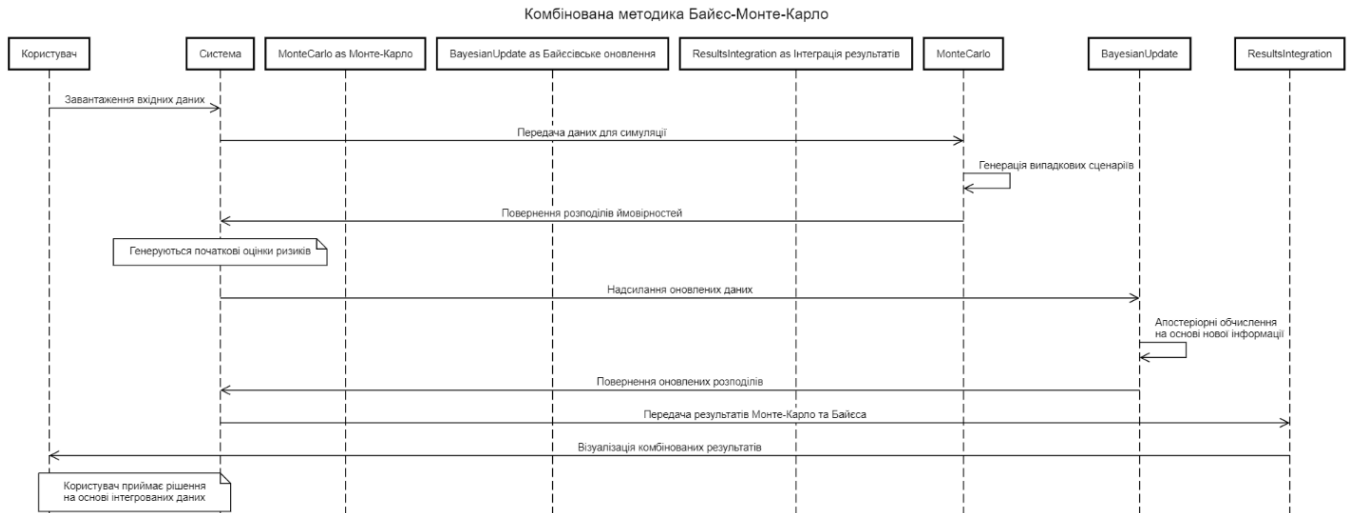


Рис. 1. Схема комбінованої методики

Висновки та перспективи.

Розроблений підхід має потенціал для впровадження в реальні IT-проекти, допомагаючи керівникам ухвалювати більш обґрунтовані рішення щодо розподілу ресурсів, управління строками та мінімізації ризиків.

Використання інтегрованих методів дозволяє досягти балансу між складністю обчислень та практичністю їх застосування. У подальшому дослідження планується розширення функціональності системи для автоматичного навчання моделей на основі отриманих даних, що сприятиме підвищенню їх ефективності.

Список використаних джерел:

1. Северінов О. В., Переметчик О. В. Комбінований метод аналізу ризиків інформаційної безпеки : thesis. 2020. URL: <http://openarchive.nure.ua/handle/document/14289> (дата звернення: 24.11.2024).
2. Тихомиров Н.П., Потравний І.М., Тихомирова Т.М. Методи аналізу та управління еколого-економічними ризиками: навч. посібник для вузів/за ред. проф. Н.П. Тихомирова. М., 2003.
3. Python Software Foundation. "Monte Carlo Simulation in Python – A Practical Guide." [Електронний ресурс]. – Режим доступу: <https://realpython.com/monte-carlo-python/>.
4. Haimes Y. Y. "Risk Modeling, Assessment, and Management." Wiley Series in Systems Engineering and Management, 2015.
5. Modarres M. "Risk Analysis in Engineering: Techniques, Tools, and Trends." CRC Press, 2016

Огляд дизайн технік, заснованих на досвіді, які використовуються при тестуванні продуктів програмного забезпечення

*Безугла Вікторія Юріївна,
студентка групи ПДМ-62,
спеціальність 121 Інженерія програмного забезпечення,
Державного університету інформаційно-комунікаційних технологій
vvv37783@gmail.com
Науковий керівник: Щербина Ірина Сергіївна,
кандидат технічних наук, доц.,
доцент кафедри Інженерії програмного забезпечення
Державного університету інформаційно-комунікаційних технологій*

У швидкоплинному світі розробки програмного забезпечення гарантування надійності та функціональності програмних продуктів має першорядне значення. Етап тестування є критично важливим компонентом життєвого циклу розробки

програмного забезпечення, спрямованим на виявлення та усунення дефектів до того, як продукт досягне кінцевого користувача.

Протягом багатьох років було розроблено та вдосконалено різноманітні техніки тест дизайну для підвищення ефективності тестування програмного забезпечення. Розглянемо дизайн техніки, засновані на досвід (experienced based), що використовують знання та ідеї, отримані з минулих проєктів і реальних кейсів. Ці методи передбачають використання евристики, дослідницького тестування та підходів на основі оцінки ризику, які дозволяють тестувальникам зосередитися на найбільш критичних і схильних до багів областях програмного забезпечення. Спираючись на свій досвід, тестувальники можуть передбачати потенційні проблеми, визначати пріоритети тестування та застосовувати перевірені стратегії для більш ефективного виявлення дефектів. Цей прагматичний підхід забезпечує надійний програмний продукт, адаптований до очікувань користувачів і галузевих стандартів.

Постановка задачі.

Провести огляд дизайн технік заснованих на досвіді, які використовуються при тестуванні продуктів програмного забезпечення.

Мета дослідження.

Покращення рівня тестування програмного забезпечення на основі проведеного огляду технік тест дизайну, заснованих на досвіді.

Результати дослідження.

Для тестування на основі досвіду використовують декілька популярних дизайн технік таких як: вгадування помилок, пошукове тестування та тестування на основі контрольного списку.

Дослідження Jorgensen, M. and Molokken-Ostvold підкреслюють цінність досвіду вгадування помилок, показуючи, що досвідчені тестувальники вправніші у визначенні проблемних областей, які автоматизовані тести можуть пропустити [1].

Дослідницьке тестування — це підхід, коли тестувальники одночасно дізнаються про програмне забезпечення, розробляють тестові сценарії та виконують їх. Цей метод

дуже адаптивний і значною мірою покладається на знання та креативність тестувальника. На відміну від тестування за сценарієм, пошукове тестування є вільною формою та дозволяє тестувальникам слідувати своїм інстинктам, щоб виявити приховані проблеми. Kaner, Bach і Pettichord стверджують, що дослідницьке тестування є особливо ефективним у ситуаціях, коли вимоги є неповними або ще тільки уточнюються, оскільки воно дозволяє більш гнучко та ретельно перевіряти програмне забезпечення. Динамічний характер цієї техніки робить її добре придатною для складних програм, орієнтованих на користувача [2].

Тестування на основі чек-лісту передбачає створення списку елементів, які потрібно перевірити на основі минулого досвіду та знання типових проблемних областей. Цей контрольний список служить посібником для забезпечення систематичної перевірки всіх критичних аспектів програмного забезпечення. Він поєднує в собі ретельність структурованого підходу з знаннями досвідчених тестувальників.

За словами Itkonen, Mäntylä та Lassenius, тестування на основі чек-листів чи списків особливо корисно для регресійного тестування, забезпечуючи, щоб жодні основні компоненти не були пропущені під час процесу тестування. Використання контрольних списків також може допомогти підтримувати послідовність і охоплення в кількох циклах тестування [3].

Висновки та перспективи.

Кожна з технік тест дизайну на основі досвіду є індивідуальною та унікальною. Для вибору підходящої слід враховувати обставини тестування, складність програмного забезпечення, кількість тестувальників, кількість часу на тестування, а також особливості вимог. Зосереджуючись на ймовірних дефектних областях, пристосовуючись до мінливих вимог і забезпечуючи повне охоплення, ці методи значно сприяють постачанню високоякісних програмних продуктів.

Список використаних джерел:

1. Jorgensen, M. and Molokken-Ostvold "Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach, and Data Analysis Method." IEEE Transactions on Software Engineering, 30(12), 993-1007 - [Електронний ресурс] – Режим доступу: <https://ieeexplore.ieee.org/document/1377193> (дата звернення: 12.06.2024).
2. Kaner, C., Bach, J., & Pettichord, B. (2001). Lessons Learned in Software Testing: A Context-Driven Approach. Wiley; 2004
3. Itkonen, J., Mäntylä, M. V., & Lassenius, C. (2007). "The Role of the Tester's Knowledge in Exploratory Software Testing." IEEE Transactions on Software Engineering, 33(5), 316-330; 2007