ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

II ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ СТУДЕНТІВ, АСПІРАНТІВ ТА МОЛОДИХ ВЧЕНИХ «ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ»

ІКСМ весна 2025

20 ТРАВНЯ 2025

KI.WUNU.EDU.UA/CONFERENCE/

ТЕРНОПІЛЬ

2025

ПРОГРАМНИЙ КОМІТЕТ КОНФЕРЕНЦІЇ

Десятнюк О.М. д.е.н., професор, ректор, Західноукраїнський національний університет
Дивак М.П. д.т.н., професор, проректор з наукової роботи, Західноукраїнський національний університет
Березький О. М. д.т.н., професор, Західноукраїнський національний університет
Антощук С.Г. д.т.н, професор, Національний університет «Одеська політехніка»
Баловсяк С.В. д.т.н, професор, Чернівецький національний університет
Бармак О.В. д.т.н., професор, Хмельницький національний університет
Батько Ю.М. к.т.н., доцент, Західноукраїнський національний університет
Винокурова О.А. д.т.н., професор, Львівський національний університет ім. Івана Франка
Возна Н.Я. д.т.н., професор, Західноукраїнський національний університет
Говорущенко Т.О. д.т.н., професор, Хмельницький національний університет
Дубчак Л.О. к.т.н., доцент, Західноукраїнський національний університет
Дунець Р.Б. д.т.н., професор, НУ "Львівська політехніка"
Ізонін І.В. к.т.н., доцент, НУ "Львівська політехніка"
Лящинський П.Б. доктор філософії з комп'ютерних наук, викладач, Західноукраїнський національний університет
Литвиненко В.І. д.т.н, професор, Херсонський національний технічний університет
Мельник Г.М. к.т.н, доцент, Західноукраїнський національний університет
Мельникова Н.І д.т.н., професор, НУ Львівська політехніка
Пелешко Д.Д. д.т.н., професор, Львівський національний університет ім. Івана Франка
Піцун О.Й. к.т.н. доцент, Західноукраїнський національний університет
Сельський П.Р. д.м.н., професор, Тернопільський національний медичний університет ім. І.Я. Горбачевського
Субботін С.О. д.т.н., професор, Національний університет «Запорізька політехніка»
Теслюк В.В. д.т.н., професор, НУ "Львівська політехніка"
Тимченко Л.І. д.т.н., професор, Державний університет інфраструктури та технологій
Цмоць І.Г. д.т.н., професор, НУ "Львівська політехніка"
Якименко І.З. к.т.н., доцент, Західноукраїнський національний університет
Яровий А.А. д.т.н., професор, Вінницький національний технічний університет
Яцків В.В. д.т.н., професор, Західноукраїнський національний університет

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ КОНФЕРЕНЦІЇ

Мельник Г.М. к.т.н., доцент, Західноукраїнський національний університет
Піцун О.Й. к.т.н., доцент, Західноукраїнський національний університет
Галунька Б.В. студент, Західноукраїнський національний університет
Фаєрчук В.В. студент, Західноукраїнський національний університет
Зінькевич О. В. студентка, Західноукраїнський національний університет
Кіт М. О. студентка, Західноукраїнський національний університет
Шевчук Ю. А. студентка, Західноукраїнський національний університет

Метою конференції є представлення та обговорення наукових і практичних результатів, сприяння активізації творчої і інноваційної діяльності студентів, аспірантів і молодих вчених.

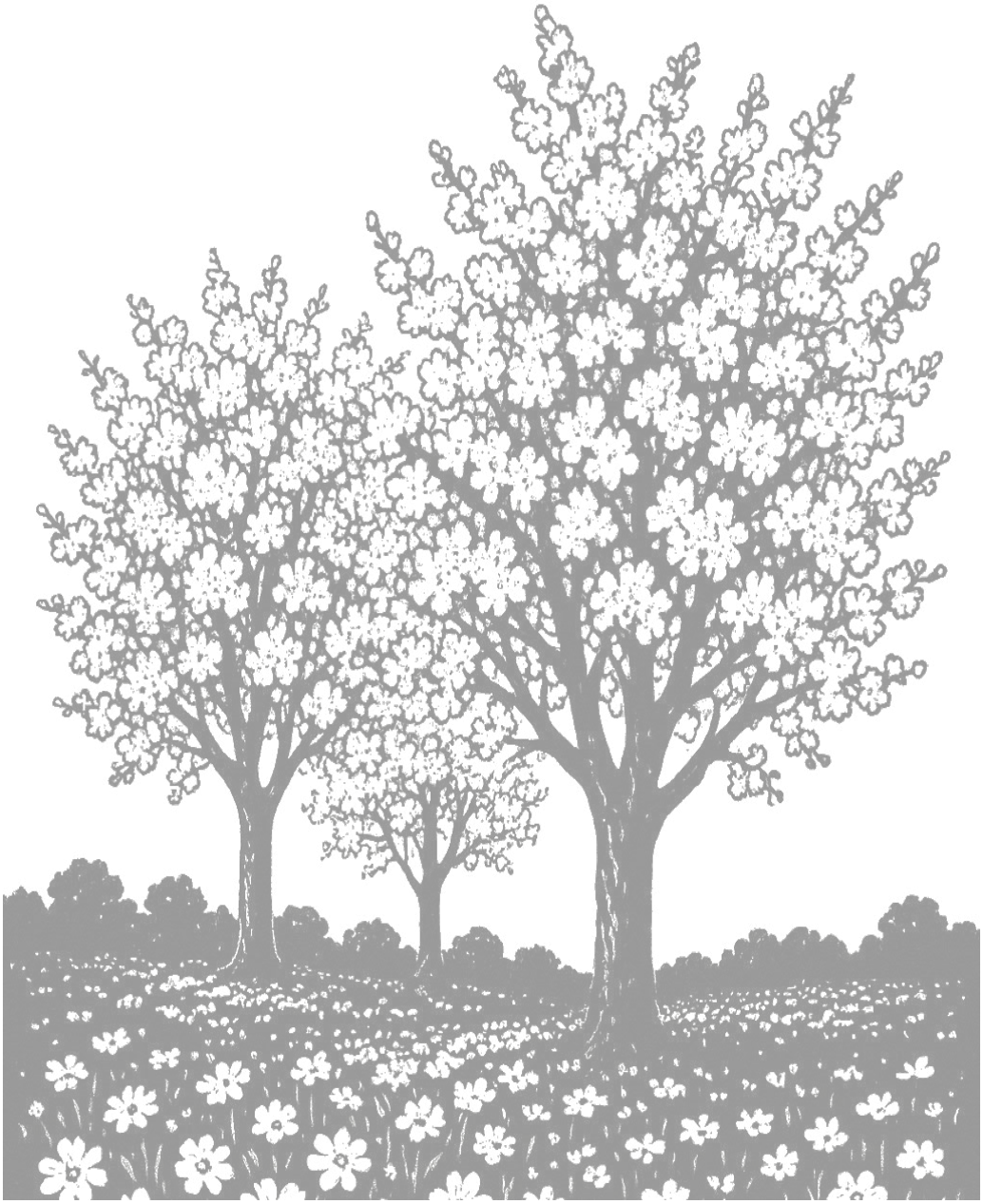
Конференція проводиться із залученням Ради Молодих Вчених та Студентського Наукового Товариства ФКІТ ЗУНУ.

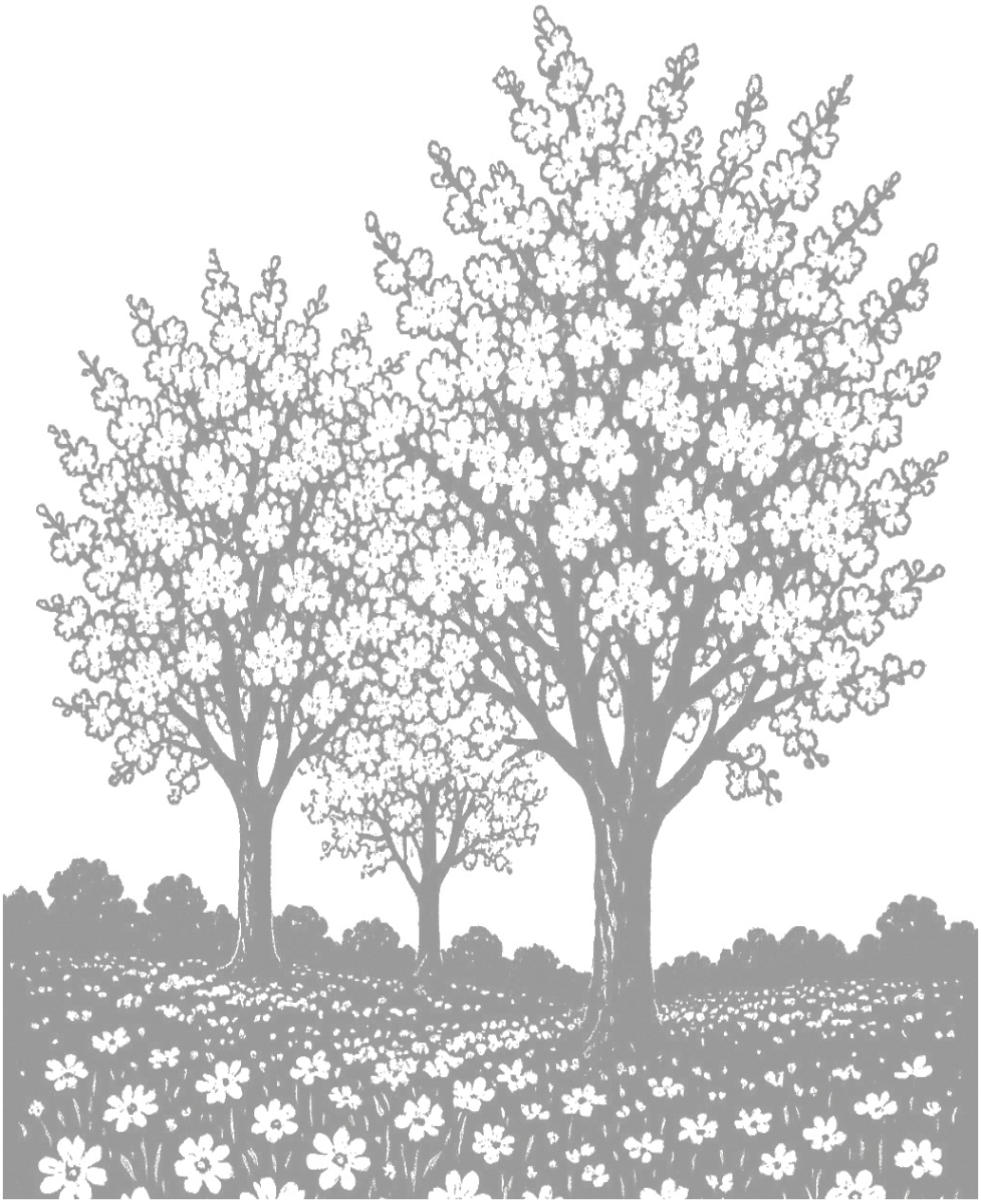
ІІ Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ весна 2025), м. Тернопіль, ЗУНУ, 20 травня 2025 р. Тернопіль, 2025

Адреса:
Вул. О. Теліги, 8, корпус №6, Тернопіль
URL: https://ki.wunu.edu.ua/conference/
e-mail: kistudconference@gmail.com

Тези доповідей подаються за оригіналом рукопису

ЗМІСТ





Фітьо Ю.В.
студентка 4 курсу ІКНІ, Національний університет "Львівська політехніка"
Науковий керівник к.ф.-м.н., доцент Сенета М.Я., кафедра АСУ
Національний університет "Львівська політехніка"

ЦИФРОВІ ТЕХНОЛОГІЇ ЯК ІНСТРУМЕНТ ДОСЛІДЖЕННЯ ТА ЗБЕРЕЖЕННЯ КУЛЬТУРНОЇ СПАДЩИНИ

Вступ. Дослідження національної культури України є важливим аспектом нашого життя. Унікальні звичаї, традиції, багатовікова історія та вишукана архітектура формують нашу ідентичність. Аналіз та збереження надбань української культури набувають великого значення, особливо під час боротьби українського народу за свою незалежність та культурні цінності. Цифрова реконструкція, 3D-моделювання, геоінформаційні системи та інші інноваційні інструменти відіграють важливу роль у документуванні стану культурних об'єктів, плануванні заходів з їхньої реставрації, а також в оцінці екологічних ризиків, спричинених воєнними діями. Зі стрімким розвитком технологій особливої популярності набувають і навчальні системи. Колись, щоб дізнатись про національну культуру, людство використовувало усну комунікацію, писемні джерела та інші носії інформації. Сьогодні всі ці матеріали можна з легкістю переглянути в цифровому форматі. Система не лише надає зручний та легкий доступ до надбань української культури, але й сприяє її ефективному вивченню за допомогою інтерактивної карти, завдань та тестів. Вона дозволить замінити друковані навчальні ресурси на цифрові. Систему культурної спадщини можна розглядати і у контексті сталого розвитку – підхід, що інтегрує охорону, використання й розвиток культурної спадщини в рамках соціального, економічного та екологічного добробуту суспільства. Така система не лише зберігає пам'ятки й традиції, а й сприяє розвитку громад, підтримує ідентичність, освіту, туризм та інновації.

Постановка задачі. Об'єктом дослідження є процес розроблення веб-системи дослідження та збереження культурних об'єктів України. Предметом роботи є методи та засоби автоматизації опрацювання національних надбань України. Метою дослідження є розроблення системи, що забезпечує ефективне збереження та представлення інформації про культурну спадщину, що сприяє її дослідженню і популяризації.

Основний матеріал. Доволі багато дослідників у своїх працях підкреслювали важливість збереження та популяризації культурної спадщини. Наприклад, автор статті [1] розповідає про те, наскільки важливо зберігати національну ідентичність за допомогою дослідження та поширення наших звичаїв, традицій, історії, мови та мистецтва. Також вона зазначає, що під час війни збереження культури стало особливо важливим, адже саме завдяки їй український народ залишається сильним та здатним протистояти ворогам, зберігаючи духовну стійкість.

Автор роботи [2] розглядає роль культурних проєктів як важливих чинників сталого розвитку. Зазначається, що в Україні тема культури в цьому контексті поки що залишається недостатньо дослідженою. У статті підкреслюється, що культурна діяльність сприяє збереженню спадщини, позитивно впливає на економіку й суспільство, формує наші цінності і поведінку, які є надзвичайно важливими для розвитку економіки й соціальної справедливості. Автор закликає розвивати культурну діяльність, яка згуртовує людей і допомагає вирішувати проблеми в суспільстві.

Відповідно до даних Міністерства культури та інформаційної політики України [3] зафіксовано значні втрати об'єктів культурної спадщини. Станом на 25 березня 2025 року було зафіксовано 1419 пошкоджених об'єктів. Найбільших втрат зазнала Харківська область – 329 пам'яток культури.

Унаслідок масштабних втрат та пошкоджень культурних об'єктів під час повномасштабного вторгнення особливої актуальності набуває збереження та дослідження культурних надбань України. Розроблена система сприятиме збереженню культурних об'єктів, підвищенню обізнаності громадян та формуванні національної ідентичності.

Існує багато сучасних освітніх систем, які автоматизують навчальний процес. Такі платформи, як: Coursera, Google Arts & Culture та інші дозволяють користувачам покращити свої знання у певній сфері, надаючи доступ до культурних ресурсів.

Платформа Coursera дозволяє користувачам вивчати різні курси: програмування, бізнес, мистецтво та ін., а також проходити тести, виконувати завдання та проєкти. Після проходження курсу користувачі можуть залишити відгук. Перевагами даної платформи є великий вибір навчальних курсів, наявність тестувань, сертифікатів, простий та зрозумілий інтерфейс. До недоліків належать складне ціноутворення та відсутність курсу дослідження культури України.

Платформа Google Arts & Culture забезпечує доступ до віртуальних музеїв, виставок та культурних об'єктів світу, а також дозволяє ознайомитись з національною культурою України, зокрема з 3D-моделями різних об'єктів. Перевагами платформи є віртуальні екскурсії, інтерактивні матеріали. До недоліків входять: відсутність перевірки знань користувачів та обмежена кількість україномовного контенту.

Аналіз платформ підтвердив важливість та актуальність розроблення веб-системи дослідження та збереження надбань української культури, що поєднує інтерактивний навчальний контент та тестування.

Система має клієнт-серверну архітектуру. Серверна частина реалізована за допомогою таких технологій, як: Java, Spring Boot та MySQL, які забезпечують високу надійність, швидкість та масштабованість системи. Клієнтська частина реалізована за допомогою React і TypeScript, які дозволяють створити зручний та інтуїтивний інтерфейс. Для відображення інтерактивної карти використовується бібліотека Leaflet, яка дає змогу досліджувати культурні об'єкти України. Вибір цих технологій забезпечив створення надійної та зручної у використанні системи.

Результатом розробки є сучасна та інтерактивна система, яка забезпечує доступ до надбань української культури. Вона надає можливість перевірки знань за допомогою інтерактивних завдань та тестів, сприяє хорошому засвоєнню навчального матеріалу та підвищує зацікавлення і залученість користувачів.

Висновки. В умовах війни цифрові технології стають не лише інструментом фіксації втрат, а й засобом опору, збереження ідентичності та підґрунтям для майбутнього відновлення. Розроблена система дослідження та збереження надбань української культури є унікальним та актуальним рішенням, адже має велике значення у формуванні національної ідентичності, підвищення рівня обізнаності та патріотизму. Також вона поєднує навчальні матеріали культурної спадщини, інтерактивні інструменти для перевірки знань та візуалізацію культурних об'єктів на карті кожного регіону України. Автоматизація збереження культури забезпечує зручний доступ до навчальних ресурсів та автоматичну перевірку знань за допомогою тестувань.

Дана система є надійним та сучасним програмним рішенням, яка була реалізована за допомогою різних сучасних технологій та засобів. Система буде корисною як для студентів та викладачів, так і музеїв та туристів. Дана розробка є перспективною, оскільки відповідає сучасним трендам цифровізації. У майбутньому система сприятиме розвитку туризму та буде мати ефективний вплив на економіку регіонів, завдяки чому культурні ресурси стануть більш доступними.

Список літератури

1. Г. А. Абрамова, "Синергетична парадигма збереження національної ідентичності через популяризацію традиційної культури". *Традиційна культура в умовах глобалізації: нові вектори розвитку: матеріали науково-практичної конференції*, Харків: О. А. Мірошниченко, 2023, С. 3–5.

2. О.В. Комарніцька. Культура та культурні проєкти як драйвери сталого розвитку. *Sciences of Europe*, No 136, 2024, С. 11–17.

3. Міністерство культури та інформаційної політики України. 1419 пам'яток культурної спадщини та 2233 об'єкти культурної інфраструктури постраждали в Україні через російську агресію. [Електронний ресурс] – Режим доступу: https://mcsc.gov.ua/news/1419-pamyatok-kulturnoyi-spadshhyny-ta-2233-obyekty-kulturnoyi-infrastruktury-postrazhdaly-v-ukrayini-cherez-rosijsku-agresiyu/ (дата звернення 11.05.2025р.).

Кришевська В.П.
студентка 4 курсу кафедри АСУ, ІКНІ, НУ ЛП
Науковий керівник к.т.н., асистент Антонів В. Я., кафедра АСУ НУ ЛП

ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ ТА АНАЛІЗУ ВИРОБНИЧИХ ДАНИХ У РЕАЛЬНОМУ ЧАСІ

Вступ. Рівень автоматизації на підприємствах та кількість впроваджених пристроїв Інтернету речей (IoT) зростають надзвичайно швидко. Причиною є прагнення підприємств підвищити ефективність виробничих і бізнес-процесів, зменшити витрати, забезпечити безперервний моніторинг обладнання та автоматизувати контроль. Водночас це створює додаткові виклики для забезпечення інформаційної безпеки та обробки великих потоків даних, які генерує велика кількість взаємопов'язаних пристроїв [1]. DDoS атаки є найбільш популярними серед кібер-злочинців для ушкодження інтернет програм або IoT систем. Вони майже миттєво виводять із ладу систему, сервер чи мережу [2].

Для контролю безпеки системи, можливість збирати, зберігати, опрацьовувати та аналізувати дані стала ключовою. Використання хмарних технологій надає можливість працювати з великими даними (Big data), з чим часто не може впоратись традиційне ПЗ аналізу даних. Хмарні рішення забезпечили масштабованість, гнучкість, безпеку та конкурентну цінову політику, прикладами є Amazon Web Services (AWS), Microsoft Azure, та Google Cloud Platform (GCP). Усі вони забезпечують доступ до низки сервісів для роботи з даними та їх аналізу [3].

Впровадження аналітичних технологій, що здатні обробляти великі об'єми даних в близькому до реального часу є необхідним та стає все більш актуальним з кожним роком, розв'язанням цієї проблеми здатна стати інформаційна система моніторингу та аналізу виробничих даних в реальному часі. Потрібно зазначити інтеграцію системи з штучним інтелектом (ШІ), для класифікації та виявлення аномалій, або іншими системами, для контролю роботи залучених IoT пристроїв, приклад – система розумного дому. Реалізація проєкту дозволить не лише підвищити ефективність управління підприємством, але й сприятиме зниженню витрат, підвищенню безпеки та якості продукції, стане важливим кроком до цифрової трансформації виробничих процесів. Це доводить актуальність створення інформаційної системи моніторингу та аналізу виробничих даних в реальному часі.

Постановка задачі. Об'єкт дослідження – процес створення інформаційної системи моніторингу та аналізу виробничих даних в реальному часі. Предмет дослідження – інформаційна система моніторингу та аналізу виробничих даних в реальному часі. Головна мета цієї роботи полягає в дослідженні найкращих шляхів реалізації інформаційної системи моніторингу та аналізу виробничих даних в реальному часі та визначенні вимог системи.

Основний матеріал. В теперішній час розробка системи що оперує петабайтами інформації потребує використання хмарних технологій, адже власна сервісна інфраструктура є невиправдано дорогою, а також ставить дані під загрозу кібер-атак. Отже проектування інформаційної системи для моніторингу виробничих даних у реальному часі потребує фундаментального розуміння архітектури хмарних платформ. Варто розглянути сервіси таких провайдерів, як AWS, Microsoft Azure та GCP. Усі вони пропонують унікальні рішення для обробки big data, проте саме рівень інтеграції з аналітичними та машинно-навчальними сервісами стає вирішальним фактором для промислових застосунків [3].

AWS вирізняється широким спектром сервісів і зрілою екосистемою для обробки великих даних, зокрема Amazon EMR, Kinesis і SageMaker. Проте складність налаштувань і порівняно висока вартість є перешкодою для швидкої інтеграції в середовищах з обмеженим бюджетом. GCP, у свою чергу, пропонує потужні інструменти аналітики, як-от BigQuery, Vertex AI та Dataflow, але дещо поступається у зрілості підтримки гібридних сценаріїв і корпоративної інтеграції, що важливо для промислових інформаційних систем.

Натомість Microsoft Azure визнано як найбільш збалансовану платформу для реалізації комплексних аналітичних рішень у промисловому середовищі. Особливо високо оцінюється інтеграція таких сервісів, як Azure Stream Analytics, Azure Databricks, Azure Event Hub та Azure

Machine Learning, що дозволяє будувати потужні пайплайни (pipeline) обробки IoT-даних у реальному часі. Крім того Azure пропонує конкурентне ціноутворення для середніх і великих обсягів даних, а також розвинуту підтримку користувачів через документацію, навчальні програми та партнерську екосистему [3].

Використання машинного навчання (ML) спричинено необхідністю швидко та якісно виявляти приховані закономірності та класифікувати аномалії у великому обсязі даних, що надходять у реальному часі. Аналіз проведений щодо існуючих рішень показав, що застосування ML-моделей дозволяє значно підвищити точність класифікації порівняно з класичними методами фільтрації. Найбільш доцільними для проєкту виявилися Python-бібліотеки для побудови моделей класифікації. Варто зазначити такі бібліотеки як Scikit-learn, Evidently AI та Pandas, вони забезпечують зручні інструменти для обробки, навчання та моніторингу якості моделей.

Вибір найбільш підходящого ML-алгоритму для побудови моделі часто є вирішальним. На основі порівняльного аналізу таких алгоритмів, як Random Forest, Support Vector Machines (SVM), Decision Trees, k-Nearest Neighbors (k-NN) і Neural Networks, які застосовуються до трафіку, зібраного з IoT-пристроїв, можна прийняти рішення. Найвищу точність при виявленні зловмисного трафіку демонструє Random Forest, водночас нейронні мережі мають перевагу при роботі з великомасштабними потоками даних [2]. Також дослідники вказують на потребу попередньої обробки та балансування даних, оскільки виробничі записи мають високий дисбаланс між “нормальним” і “злоякісним” трафіком.

Для забезпечення контролю якості роботи моделей машинного навчання в динамічному середовищі передбачається використання спеціалізованого інструменту Evidently AI [4]. Це сучасне рішення у сфері MLOps, яке дозволяє здійснювати перевірку стабільності та продуктивності моделей у виробничому середовищі. Evidently AI надає функції виявлення змін Data Drift і Concept Drift, тобто у розподілі вхідних даних і в логіці цільової змінної. Такий моніторинг особливо важливий у системах, які працюють на потоці реального часу, адже дозволяє вчасно виявляти збої в роботі моделі або зміни в структурі даних, на які необхідно реагувати, наприклад, перенавчанням або корекцією моделі.

Одним із основних компонентів ефективного аналізу виробничих даних у реальному часі є візуалізація результатів моніторингу та аналітики. Сучасні аналітичні системи, які працюють з потоками даних з IoT-пристроїв, зобов’язані не лише обробляти інформацію, а й представити її у зручному, зрозумілому вигляді для користувача. Головною метою інтерфейсу є миттєва передача критичної інформації, що дозволяє оператору вчасно реагувати на інциденти. Візуальна складова значно зменшує когнітивне навантаження [5]. Важливим є і перегляд історичних подій, для аналізу причин відхилень, а також інтуїтивний дизайн, який адаптується до різних сценаріїв.

Для побудови дашбордів та графіків використано Power BI, який добре інтегрується з Microsoft Azure, забезпечує можливість створення орієнтованих на користувачів звітів і графіків, які відображають найважливіші параметри стану даних та динаміку подій у реальному часі. Крім того, система передбачає підказки, які на основі отриманих даних та ML-аналізу пропонують користувачу попередні висновки та можливі шляхи подальших дій, наприклад повідомлення про ймовірну DDoS атаку та рекомендації для оператора. Цей підхід дозволяє виявляти потенційні загрози і здійснювати рішення, що підвищують рівень захищеності даних виробництва.

Отже, було визначено структуру інформаційної системи моніторингу та аналізу виробничих даних у реальному часі. Система допоможе легко та швидко помічати зміни в поведінці IoT пристроїв, підвищуючи рівень автоматизації, безпеки та продуктивності підприємств.

Функціональні вимоги до розроблюваної інформаційної системи:

- прийом даних з IoT-пристроїв;
- обробка потокових даних у режимі реального часу;
- збереження даних у хмарному сховищі даних для подальшого аналізу;
- виявлення відхилень за допомогою машинного навчання;

- порівняння поточних показників із історичними даними;
- формування аналітичних звітів та візуалізацій у вигляді графіків та дашбордів;

Також на рисунку 1 наведено структурну схему системи, вона дозволяє наочно побачити взаємодію всіх елементів та формати в яких передаються дані.

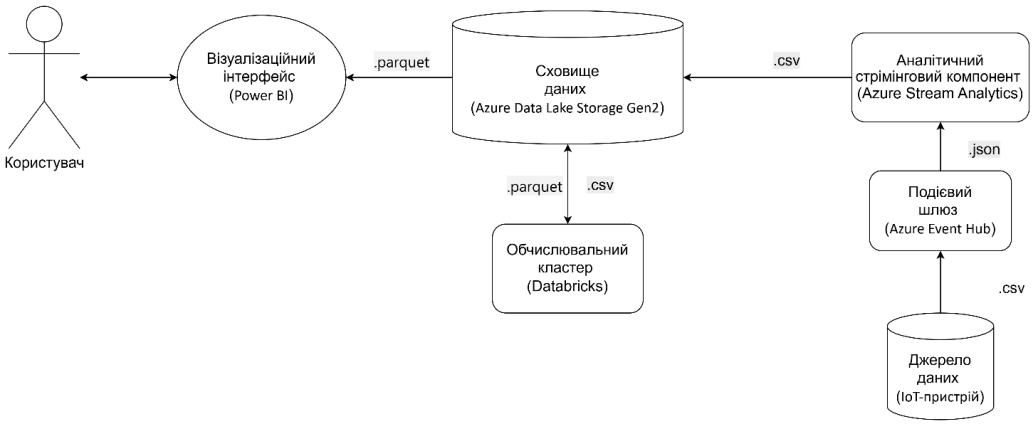


Рисунок 1 – Структурна схема системи

Можна побачити, що з IoT-пристрою надходять сирі дані у режимі реального часу, подієвий шлюз (Azure Event Hub) приймає дані від IoT і аналітичний стрімінговий компонент отримує дані з Event Hub та зберігає їх в .csv форматі в Azure Data Lake Storage Gen2. Сховище даних забезпечує довготривале, масштабоване збереження історичних даних у вигляді файлів, а також вже оброблених даних. Обчислювальний кластер Databricks – основна платформа для складної обробки даних, аналізу та реалізації моделей машинного навчання. Результати з Databricks записуються назад у хмарне сховище, а також передаються до візуалізації. Power BI – кінцевий інтерфейс для користувачів, які працюють з системою в реальному часі.

Висновки. Було розроблено структуру інформаційної системи моніторингу та аналізу виробничих даних у реальному часі. Платформою для реалізації визначено Microsoft Azure, адже вона є найбільш збалансованою, порівняно з AWS та GCP, разом з використанням алгоритму машинного навчання Random Forest та візуалізацією, що відповідає рекомендаціям, система є основним конкурентом у галузі. Завдяки проведеному аналізу, рішення здатне своєчасно виявляти аномалії та потенційні атаки, що дозволяє запобігати збиткам, а також є гнучким та легко адаптується до потреб різних компаній.

Система є необхідною та незамінною частиною роботи кожного підприємства, що працює з IoT-пристроями. Її особливостями є здатність забезпечити якісний аналіз великих даних з використанням ШІ, хмарних технологій та провідних рішень візуалізації, що надає можливість підвищити ефективність, безпеку та контроль якості роботи підприємства.

Список літератури

1. Imane Kerrakchou, Adil Abou El Hassan, Sara Chadli, Mohamed Emharraf, Mohammed Saber, “Selection of efficient machine learning algorithm on Bot-IoT dataset for intrusion detection in internet of things networks” *Indonesian Journal of Electrical Engineering and Computer Science* Vol. 31, No. 3, P. 1784-1793, 2023. DOI: 10.11591/ijeecs.v31.i3

2. Yee Zi Wei, Marina Md-Arshad, Adlina Abdul Samad, Norafida Ithnin “Comparing Malware Attack Detection using MachineLearning Techniques in IoT Network Traffic” *International Journal of Innovative Computing*, 13(1), P. 21– 27, 2023. DOI: 10.11113/ijic.v13n1.384

3. Samon Daniel, Seraphina Brightwood, Joseph Oluwaseyi, “Cloud-based big data analytics (aws, azure, google cloud)”, 2024. Доступ: https://www.researchgate.net/publication

4. Y. Swathi, Manoj Challa “From Deployment to Drift: A Comprehensive Approach to ML Model Monitoring with Evidently AI” *Recent Advances in Signals and Systems*, P. 307-320, 2024.

5. Oluwaseyi Ayotunde Akano, Enobong Hanson, Chukwuebuka Nwakile and Andrew Emuobosa Esiri “Designing real-time safety monitoring dashboards for industrial operations: A data-driven approach” *Global Journal of Research in Science and* Technology, 02(02), P. 001–009, 2024.

Остапчук С.М.[1], Домбровський В.І.[2]
[1] студент 4 курсу ФКІТ ЗУНУ, аспірант ФКІТ ЗУНУ
Науковий керівник: к.т.н., доцент Дубчак Л.О., кафедра КІ ЗУНУ

СУЧАСНИЙ СТАН ВІДНОВЛЮВАНОЇ ЕНЕРГЕТИКИ В УКРАЇНІ ТА СВІТІ: КЛЮЧОВІ ТЕНДЕНЦІЇ

Вступ. В умовах глобальних кліматичних змін, зростання споживання енергії та вичерпності традиційних джерел пального, відновлювана енергетика набуває дедалі більшого значення як стратегічний напрям розвитку світової енергетики. Сучасні технології у сфері сонячної, вітрової, біоенергетики та гідроенергетики дозволяють забезпечувати потреби в електроенергії з мінімальним впливом на довкілля, знижуючи викиди парникових газів та залежність від імпорту викопного палива. Впродовж останніх п’яти років світ демонструє стабільне зростання інвестицій у відновлювану енергетику, хоча геополітичні, економічні та соціальні виклики суттєво впливають на темпи та особливості цього процесу в окремих регіонах. Україна, перебуваючи в умовах війни, одночасно стикається із значними викликами й демонструє стійкість у реалізації національних планів енергетичної трансформації [1].

Постановка задачі. Об’єкт дослідження – відновлювана енергетика України та світу. Предмет дослідження – технології сонячної, вітрової, біоенергетики та гідроенергетики. Метою цього дослідження є аналіз сучасного стану розвитку відновлюваної енергетики як у глобальному контексті, так і в Україні, з урахуванням останніх статистичних даних, досліджень та офіційних стратегій. Особлива увага приділяється перспективам галузі, тенденціям розвитку, викликам та можливим сценаріям майбутнього.

Основний матеріал. У період з 2020 по 2025 рік спостерігається безпрецедентне зростання ролі відновлюваної енергетики у світовій енергетичній системі. За даними Міжнародного енергетичного агентства (IEA), відновлювані джерела енергії забезпечують понад 95% приросту нових електрогенеруючих потужностей у світі. Очікується, що вже до 2025 року ВДЕ стануть провідним джерелом електроенергії, випередивши вугільну та газову генерацію [2, 3]. Це зростання стало можливим завдяки розвитку технологій, зниженню вартості обладнання та посиленню кліматичних зобов’язань країн.

Азіатсько-Тихоокеанський регіон наразі є лідером за темпами впровадження ВДЕ, формуючи майже половину (47,7%) світового ринку у 2023 році. Значну роль відіграє Китай, де державна політика спрямована на масштабне впровадження сонячних та вітрових електростанцій. Згідно з аналітикою Financial Times, Китай до 2028 року зможе виробляти до 50% електроенергії з низьковуглецевих джерел, що суттєво змінює глобальну енергетичну карту та зміцнює його позиції як нового лідера у сфері чистої енергії [2, 4, 5].

Водночас розвинені країни, зокрема Європа та США, зіштовхуються з новими викликами. Економічна нестабільність, інфляція та високі відсоткові ставки призвели до скасування або відтермінування низки масштабних проектів у сфері ВДЕ. Крім того, у США політичні дебати навколо зеленої трансформації можуть негативно впливати на реалізацію національних кліматичних цілей. Такі внутрішні суперечності ставлять під загрозу досягнення ключових орієнтирів у сфері декарбонізації.

Зовсім інша ситуація спостерігається у країнах Латинської Америки, де поширюються ініціативи громадської енергетики. У віддалених регіонах, зокрема в Амазонії, мешканці самостійно встановлюють сонячні панелі, забезпечуючи себе електроенергією без централізованої підтримки. Цей тренд не лише сприяє підвищенню енергетичної незалежності громад, а й демонструє новий підхід до енергетичного суверенітету знизу вверх [6].

До початку повномасштабної війни у 2022 році Україна демонструвала стабільний розвиток у сфері відновлюваної енергетики. Зокрема, за даними UkraineInvest, на той момент в країні було встановлено понад 10 ГВт потужностей ВДЕ, з яких значна частина припадала на сонячну та вітрову генерацію. Україна активно інтегрувала ВДЕ у свою енергетичну систему, впроваджуючи відповідне законодавство та залучаючи інвесторів через систему "зеленого тарифу" [7-9]. Приблизно чверть потужностей, зокрема об'єкти в південних і східних регіонах

країни, опинилися на тимчасово окупованих територіях або зазнали пошкоджень внаслідок бойових дій. Це створило нові виклики для енергетичної безпеки України та змусило уряд і бізнес шукати нові підходи до відновлення і розвитку галузі. У 2023 році уряд України затвердив нову енергетичну стратегію, яка передбачає досягнення 27% частки ВДЕ у загальному кінцевому споживанні енергії до 2030 року. Ця мета відповідає курсу України на євроінтеграцію та узгоджується з політикою ЄС у сфері декарбонізації [2, 3].

Попри складну ситуацію, в країну продовжують надходити інвестиції у відновлювану енергетику. Так, у 2023 році в сонячну енергетику було інвестовано близько 150 млн доларів, а також введено в експлуатацію Тилігульську вітрову електростанцію потужністю 114 МВт. Ці проєкти свідчать про довіру інвесторів до потенціалу українського ринку.

Важливим кроком стало реформування ринку підтримки ВДЕ. Замість фіксованих "зелених тарифів" Україна поступово переходить до конкурентної аукціонної моделі та впроваджує механізм контрактів на різницю. Це має забезпечити прозорість, зменшення фінансового навантаження на споживачів і стимулювання здорової конкуренції на ринку.

Висновки. Сучасна відновлювана енергетика переживає період активного росту, трансформації та стратегічного переосмислення. У глобальному масштабі спостерігається стійка тенденція до зниження залежності від викопного палива, що обумовлена як економічними факторами, так і потребами боротьби зі зміною клімату. Впровадження новітніх технологій, розвиток систем накопичення енергії, водневої інфраструктури та «розумних» мереж, а також поступова децентралізація енергетичних систем формують нову архітектуру світової енергетики. Україна, попри війну та виклики, продовжує демонструвати рішучість у розвитку відновлюваної енергетики, інтегруючи європейські стандарти, реформуючи регуляторне поле та залучаючи інвестиції навіть у складних умовах. Енергетична стратегія країни до 2030 року, орієнтована на збільшення частки ВДЕ та декарбонізацію, свідчить про намагання не лише подолати кризу, а й використати її як стимул для модернізації та переходу до сталої моделі розвитку. Разом з тим, ефективне розширення ВДЕ потребує комплексного підходу, який охоплює як технічні інновації, так і соціально-економічну адаптацію, модернізацію інфраструктури, створення стимулюючого середовища для інвесторів та активну участь місцевих громад. Усе це визначає відновлювану енергетику не лише як джерело чистої енергії, а як ключовий елемент формування енергетичної безпеки, економічної стійкості та екологічного майбутнього України й світу.

Список літератури

1. IEA. Renewables 2020 – Analysis. https://www.iea.org/reports/renewables-2024
2. Market Publishers. Renewable Energy Global Industry Guide 2019-2028. https://marketpublishers.com
3. UkraineInvest. Renewable energy. https://ukraineinvest.gov.ua
4. Галина Шмідт: Відновлювана енергетика має стати головною генерацією у світі (2022). URL: https://ua-energy.org/uk/posts/halyna-shmidt-vidnovliuvana-enerhetyka-maie-staty-holovnoiu-heneratsiieiu-v-sviti6.
5. Renewable capacity statistics 2024. URL: https://www.irena.org/Publications/2024/Mar/Renewable-capacity-statistics-20247.
6. Альтернативна енергетика в Україні: актуальний стан (2024). URL: https://dlf.ua/ua/alternativna-energetika-v-ukrayini-aktualnij-stan/
7. Dubchak L. Modern renewable energy sources and methods for detecting their defects. Computer systems and information technologies, V.2, 2024, pp.21-26.
8. Dubchak, L.; Sachenko, A.; Bodyanskiy, Y.; Wolff, C.; Vasylkiv, N.; Brukhanskyi, R.; Kochan, V. Adaptive neuro-fuzzy system for detection of wind turbine blade defects. *Energies* 2024, *17*, 6456.
9. Roga, S.; Bardhan, S.; Kumar, Y.; Dubey, S.K. Recent technology and challenges of wind energy generation: A review. *Sustain. Energy Technol. Assess.* 2022, *52*, 102239.

Панчук О. В.
студент 4-го курсу кафедра АСУ, Національний університет "Львівська політехніка"
Науковий керівник к.т.н., доцент Скорохода О. В. кафедра АСУ
Національний університет "Львівська політехніка"

ІНТЕЛЕКТУАЛЬНА СИСТЕМА БАГАТОРІВНЕВОЇ МОДЕРАЦІЇ КОНТЕНТУ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Вступ. Забезпечення безпечного та цивілізованого онлайн-середовища вимагає ефективної модерації користувацького контенту. У міру зростання обсягів інформації, що генерується користувачами, традиційна ручна модерація стає недостатньою – більшість платформ звертається до рішень на базі штучного інтелекту (ШІ). Сучасні алгоритми здатні автоматично виявляти образливі висловлювання, пропаганду ненависті, порнографію, сцени насильства та інші порушення в текстових, візуальних та аудіоматеріалах. Втім, при застосування ШІ у модерації контенту постають певні викликами пов'язані з забезпечення точності, масштабованості і врахування контексту.

Постановка задачі. Об'єкт дослідження – процес модерації користувацького контенту в онлайн-системах. Предмет дослідження – методи та засоби штучного інтелекту для мультимодальної автоматичної модерації, архітектурні рішення багаторівневих модераційних систем. Головною метою є дослідження існуючих рішень модерації контенту та розробка концепції універсального агента штучного інтелекту для автоматизованої багаторівневої модерації користувацького контенту, який забезпечуватиме інтегровану обробку текстів, зображень та аудіо, комбінує правила і моделі глибокого навчання, і перевищує за основними показниками точності та продуктивності відомі на сьогодні системи.

Основний матеріал. Розглянемо існуючі рішення для модерування контенту. Наразі домінують моделі на основі обробки природної мови. Сервіс Perspective API (Jigsaw/Google) оцінює токсичність коментарів і працює в реальному часі, підтримуючи 18 мов, проте обмежується переважно виявленням образливої лексики (toxicity, insult, threat). OpenAI пропонує Moderation API на основі своєї великої мовної моделі, що класифікує текст по 6 категоріях небезпеки [1]. Дослідження показують, що великі моделі (GPT-4) перевершують класичні підходи у точності виявлення hate speech. Microsoft Azure Content Safety підтримує багатомовну модерацію текстів (100+ мов) та інтегрує людину для перевірки сумнівних випадків. Рішення, які є у відкритому доступі, включають моделі типу BERT, RoBERTa, XLM-R, тонко налаштовані на наборах даних токсичних коментарів – вони доступні через HuggingFace. У практиці часто застосовують комбінацію правил (словники ненормативної лексики) та моделей глибокого навчання, аби охопити як явні, так і завуальовані порушення.

Для модерації зображень існують комерційні сервіси (Google Vision SafeSearch, Amazon Rekognition, Azure Content Safety) на основі CNN/Vision Transformer мереж розпізнають порнографію, насильство, кров, зброю і т.д., повертаючи відповідні мітки з оцінками впевненості. Наприклад, Rekognition визначає до 8 категорій небажаного вмісту з достовірністю >95% у багатьох випадках. Спеціалізовані API (Hive, Sightengine) дозволяють мультимодальну модерацію (зображення, відео, GIF) і надають гнучке налаштування під потреби споживача. Окрема категорія – методи перцептивного хешування (Microsoft PhotoDNA) для виявлення точно відомих заборонених зображень (наприклад, екстремістські символи) шляхом порівняння зі спектральною базою хешів [2]. Поєднання нейромережевого аналізу з хеш-фільтром є найпотужнішим рішенням та використовується такими гігантами як Facebook та Google.

В сфері модерації аудіо рішення базуються на автоматичному розпізнаванні мовлення (ASR) з подальшою модерацією отриманого тексту. Популярні платформи (AssemblyAI, Sieve) використовують алгоритм, в якому спочатку відбувається обробка аудіо, далі йде транскрипція (наприклад, моделлю Whisper) та на останньому етапі аналіз тексту NLP-моделлю [3]. Це дозволяє виявляти образи, ненависть, погрози, згадки заборонених речовин, самогубства тощо, які прозвучали у голосовому чаті або подкасті. Ключовим фактором є якість розпізнавання: сучасні моделі (Whisper, Google Speech) досягають <10% помилок, і навіть нецензурна лексика

розпізнається (якщо не ввімкнено фільтр цензурування). Проведемо аналіз кількох найбільш популярних сервісів для модерування коненту.

Найбільш дієвими інструменти для тексту виявилися моделі GPT-3/4 (OpenAI) та новітні багато-мовні трансформери (як Perspective API), які продемонстрували найвищі бали точності [4]. Для зображень – спеціалізовані глибокі CNN, навчені на порнографії і насильстві (наприклад, EfficientNet на датасеті NP3m), у поєднанні з PhotoDNA для відомого контенту. Для аудіо найкраще себе показали Whisper (ASR) + RoBERTa Toxic (NLP).

Незважаючи на досить високі показники точності існуючих рішень, вони мають чимало недоліків та обмежень. Пропоную побудувати концепцію універсального агента штучного інтелекту, який зможе здійснювати багаторівневу модерацію мультимодального контенту. Він поєднуватиме детерміністичні фільтри, навчені моделі та комплексний підхід до прийняття остаточного рішення за рахунок об'єднання результатів з різних шарів перевірки. Архітектура системи включатиме три основні шари.

Попередня фільтрація (перший рівень) – перевірятиме контент на наявність очевидних заборонених шаблонів. Для тексту – пошук у чорному списку слів/фраз (ненормативна лексика, расові образи, екстремістські гасла). Для зображень – порівняння гешів (наприклад, PDQ) зі списком відомих нелегальних зображень; перевірка метаданих. Для аудіо – аналіз спектру на наявність заглушених "beep" (ознака цензурованої лайки) або використання швидкого ASR для ключових слів. Контент, який не проходитиме цей фільтр, буде блокуватися або позначатиметься для ретельної перевірки вручну (в залежності від категорії). Цей модуль буде реалізований простими та швидкими алгоритмами, які забезпечать початковий відсів.

Аналіз контенту з використанням ШІ (другий рівень) – основний інтелектуальний блок, що складатиметься з підмодулів для кожного типу контенту. Тут використовуватиметься NLP-модуль – глибока нейромережа, яка класифікуватиме текст за кількома категоріями: образа/булінг, мова ненависті, погрози, заклики до суїциду, сексуальний контент, спам, тощо. За необхідності, перед аналізом текст можна нормалізувати (провести видалення зайвих символів, розшифрувати скорочення). NLP-модуль також використовуватиметься для аналізу тексту, витягнутого з зображень (через OCR) та транскрибованого з аудіо – таким чином, уніфікується обробка текстової інформації незалежно від джерела.

На другому рівнів також відбуватиметься перевірка зображень за допомогою CV-модулів. Конволюційна нейронна мережа або трансформер визначатиме наявність на зображенні забороненого вмісту. Модуль може давати декілька виходів (бінарних або з градацією) по категоріях: оголеність, сексуальні дії, насильство, кров і жорстокість, ворожа символіка, наркотики/алкоголь, тощо. Для відео модуль аналізуватиме окремі кадри (наприклад, 1 кадр кожні 2 секунди). Якщо на зображенні є текст (плакати, меми) – викликатиметься OCR і отриманий текст надсилатиметься до NLP-модуля. CV-модель буде поєднуватися зі спеціалізованими детекторами: розпізнавання конкретних логотипів чи символів (до прикладу, свастика, прапори терористичних організацій), визначення сцен насильства (бійка, застосування зброї). Це покращуватиме повноту аналізу.

До другого рівня відноситься звуковий модуль, який виконуватиме перетворення аудіо в текст за допомогою ASR. Використовуватиметься сучасна модель (наприклад, Whisper, яка підтримує багато мов і стійка до сленгу). Отриманий текст сегментуватитиметься (за паузами або репліками) і кожен сегмент буде передаватися в NLP-модуль для модерації змісту. Паралельно аудіо може бути проаналізоване на нетривіальні мовні сигнали: детектор різкого підвищення гучності (крик), детектор специфічних звуків (постріли, вибух) – якщо такі присутні, вони слугують додатковими індикаторами (наприклад, крик та нецензурна лексика означатиме ознаку агресивного контексту).

Об'єднання результатів та прийняття рішення (третій рівень). Результати з усіх підмодулів другого рівня агрегуються у централізованому модулі прийняття рішень (Rule Engine). Цей модуль матиме певний набір правил та логіки, що враховуватиме результати моделей (ймовірності по категоріях) і метадані (про користувача, контекст). На даному рівні існує перелік правил. Які дії вчинити при певних комбінаціях результатів. Наприклад, "Якщо токсичність тексту >0.9 або виявлено серйозну погрозу – автоматично відхилити/приховати повідомлення".

Пороги впевненості для різних категорій. Наприклад, допускаємо певний рівень оголеності (до 0.5) без блокування, але якщо >0.5 і <0.8 – потрібна ручна перевірка, якщо >0.8 – блокуємо одразу. Тут також важливою є синергія ознак. Наприклад, якщо зображення саме по собі сумнівне (легка оголеність 0.5) і текст під ним теж (дещо непристойний жарт 0.6 токсичності), то в сукупності вирішуємо, що пост треба обмежити (хоча окремо кожен компонент міг би пройти). Це дозволятиме врахувати контекст між різними типами контенту. Модуль прийняття рішення виноситиме вердикт для контенту: «допустимий», «неприйнятний» або «потребує додаткового розгляду». У разі виявлення неприйнятного контенту може виконуватись і автоматична дія (видалення посту, блокування користувача, відправка попередження).

Висновки. Розроблено концепцію універсального модераційного агента з багаторівневою обробкою різних видів контенту, яка повинна забезпечити високу точність і масштабованість. Агент перевершує наявні рішення завдяки адаптивності, гнучкості та покриттю всіх форматів контенту.

Список літератури

1. Content Moderation: What It Is, How It Works, and the Best APIs. *AssemblyAI | AI models to transcribe and understand speech*. URL: https://www.assemblyai.com/blog/content-moderation-what-it-is-how-it-works-best-apis-2 (дата звернення: 11.05.2025).
2. Top CSAM Detection Tools: A Comprehensive Guide to Protecting Your Platform. *In-app Chat SDK & API For Messaging And Calling - CometChat*. URL: https://www.cometchat.com/blog/csam-detection-tools (дата звернення: 02.05.2025).
3. AI Audio Moderation: How it works and a complete implementation guide. URL: https://www.sievedata.com/resources/how-to-perform-ai-audio-moderation (дата звернення: 06.05.2025).
4. Digital Guardians: Can GPT-4, Perspective API, and Moderation API reliably detect hate speech in reader comments of German online newspapers? Disclaimer: This research aims to combat hate speech and, therefore, contains examples of hate speech or offensive language, for analysis and educational purposes. URL: https://arxiv.org/html/2501.01256v1#abstract (дата звернення: 09.05.2025).

Голубець Т.Р.
аспірант 1 курсу ІКНІ НУ «Львівська Політехніка»
Науковий керівник д.т.н. Медиковський М.О., ІКНІ НУ «Львівська Політехніка»

ВИЗНАЧЕННЯ КЛЮЧОВИХ ФАКТОРІВ ПРИ ПЕРЕДБАЧЕННІ РІШЕННЯ ЛЮДИНИ ПРО ВИЗНАННЯ ВТРАТИ КОНТРОЛЮ НАД ПРОЦЕСОМ.

Вступ. Вивчення особливостей прийняття рішень людиною є важливою частиною різних наук, в першу чергу психології чи соціології. В той же час актуальність взаємодії людей з комп'ютерними системами при потребі здійснити вибір продовжує зростати. За кілька останніх десятиліть системи пошуку інформації, сповіщення, представлення даних стають все популярнішими. Всі вони здобувають визнання через те, що доводять власну ефективність. Розробка таких проектів, які об'єднано можна назвати системами прийняття рішень, повинна здійснюватися на основі розуміння двох факторів: загального процесу прийняття рішення людиною і впливу різних типів допоміжних автоматизованих засобів на цю операцію.

Проведене дослідження спрямоване на визначення рівня впливу загальних факторів на прийняття рішення людиною і прогнозування результату - обраного варіанту. Особливо активно питання здійснення вибору досліджується у економічній сфері, де важливо зрозуміти як людина сприймає товар чи послугу залежно від подачі, кількості альтернатив, різниці між цінами [1]. Дослідження в галузі психології чи когнітивної нейронауки вивчають процес прийняття рішення з іншого боку - вони приділяють більше уваги емоціям, активності різних частин мозку, самопочуттю людини [2].

Фактори, що впливають на прийняття рішення залежать від конкретного завдання, проте досліджень з аналізом цих чинників у різних галузях поки зовсім небагато. Наприклад, ми не можемо відповісти точно, що впливає на абстрактного диспетчера. Авіадиспетчер чи оператор АЕС враховують у своїй роботі абсолютно різні параметри, але ми можемо виділити спільне: досвід, час на прийняття рішення та складність ситуації. Незважаючи на це звуження, завдання залишається надто широким, тому це дослідження стосується одного з критичних випадків у прийнятті рішення - визнання втрати контролю над процесами.

Розуміння не лише переліку факторів, що відіграють роль при прийнятті рішень, але й рівня їх впливу може бути корисним і конкретним працівникам, перед якими постає вибір, і їх керівникам. Також висновки цього дослідження можуть бути враховані при розробленні автоматизованих систем, що не просто надають інформацію, а здатні адаптуватися до особливостей оператора, який буде з ними взаємодіяти.

Для аналізу рішень людини використано дані шахових поєдинків. У цьому дослідженні події у межах партії та показники, що характерні для шахів, зведено до загальніших понять для можливості перенесення результатів у будь-яку сферу, де людина приймає певні рішення. Таким чином визнання поразки у грі розглядається як визнання втрати контролю чи визнання безнадійності ситуації, досвід та рівень навичок визначається рейтинговим показником ELO [3,4], а складність ситуації - оцінкою шансів на перемогу комп'ютерним рушієм Stockfish [3,5].

У дослідженні передбачається, що фактори, які є важливими для методів машинного навчання, є значущими і в процесі, який досліджується - прийняття рішення людиною.

Постановка задачі. Об'єктом дослідження є процес прийняття рішення людиною про визнання втрати контролю над ситуацією. Предмет дослідження - фактори, що мають вирішальне значення у визнанні поразки, та методи прогнозування таких людських рішень. Метою даної роботи є розроблення ефективних методів і засобів для прогнозування рішення людини про втрату контролю над ситуацією та аналіз значущості факторів. Дослідження охоплює обробку даних для визначення переліку значущих вхідних параметрів, статистичний аналіз даних, створення та навчання моделей для прогнозування вибору людини про визнання втрати контролю.

Основний матеріал. Обробка набору даних про шахові партії полягала у формуванні параметрів, що описують не цілий матч, а конкретну ситуацію після одного з ходів. У процесі

такого переходу проведено також умовний поділ параметрів на три групи: досвід, часове обмеження та складність ситуації. Таким чином рівень впливу кожної з груп, що визначено у результаті дослідження, є об'єднаним показником важливості кількох параметрів, які до неї входять.

Після очищення даних та зміни їх формату для опису ситуацій набір параметрів складається з:

- досвід: рейтинговий показник ELO гравця, що приймає рішення здатися, та його опонента;
- часове обмеження: відношення часу, що залишився у шахіста, до початкового обмеження на початку партії та ідентичний показник опонента;
- складність ситуації: 5 останніх оцінок шансів гравця на перемогу на основі положення фігур на дошці. Шанси гравця на перемогу визначаються завдяки комп'ютерному рушію Stockfish;
- додатковий параметр: кількість ходів в партії до виникнення ситуації, що розглядається.

Таким чином, досвід та часове обмеження характеризуються парами параметрів: один описує гравця, а інший - опонента. Останні оцінки шансів гравця на перемогу демонструють складність ситуації, проте не лише в моменті, але й дають певний контекст зміни обставин протягом п'яти ходів, що передували вибору шахіста між визнанням поразки та продовженням боротьби.

Визначення кореляційних зв'язків між рішенням визнати поразку та вхідними параметрами показало залежність, в першу чергу, від особливостей положення фігур на дошці. Найвищі коефіцієнти рангової кореляції Спірмена визначено у оцінок складності ситуацій (0.58 - 0.78) та часового обмеження (0.48, 0.5). Результати використання цього методу демонструють, що рішення про визнання втрати контролю мало залежить від досвіду. Тим не менш, певна закономірність існує - за п'ять ходів до визнання поразки у більш досвідчених гравців ситуація на 39% краще від ідентичного показника у початківців. Професіонали здатні довше не доводити ситуацію до критичної, проте швидше визнають втрату контролю, якщо позиція справді безнадійна.

Також, коефіцієнти кореляції Спірмена отримані на основі набору даних про партії шахістів з досвідом є суттєво вищими, за ідентичні показники визначені на іграх початківців. Наприклад, вхідні параметри, що описують часове обмеження, корелюють з рішенням про визнання поразки в середньому на 0.16, а параметри складності ситуації - на 0.23, якщо розглядати поєдинки аматорів (до 500 ELO). Для партій більш досвідчених гравців (від 2000 ELO) ці показники становлять 0.38 та 0.66 відповідно. Це може свідчити про те, що професіонали діють більш раціонально, в їх поведінці простіше виявляти закономірності, а, відповідно, і передбачати дії.

Прогнозування рішень людини про визнання втрати контролю здійснено з використанням методів машинного навчання: глибокої нейронної мережі, випадкового лісу та градієнтного бустингу XGBoost. Навчання проведено на основі 8 мільйонів ситуацій з шахових партій [3]. Збалансовані тренувальні дані забезпечено завдяки вибору не більше 2 ситуацій з кожної партії, а також однаковою кількістю ситуацій, де людина прийняла рішення визнати втрату контролю, та тих, де продовжила гру. Завдяки такому підходу точність та повнота є високою при прогнозуванні обох класів.

На основі сформованого набору даних проведено навчання моделей, кожна з яких продемонструвала точність більше 90%:

- нейронна мережа - 90.8%;
- випадковий ліс - 91.4%;
- градієнтний бустинг XGBoost - 91.3%.

Навчені моделі дозволяють не лише спрогнозувати рішення людини в певній ситуації, а й визначити, наявність яких вхідних параметрів має найбільш критичний вплив на точність прогнозування.

На основі навчених моделей визначено втрату точності при перестановці значень кожного параметра випадковим чином. В результаті визначення середнього між показниками різних методів машинного навчання виявлено, що найбільш впливовими є остання (0.147) та передостання (0.135) оцінка складності ситуації. Через меншу важливість оцінок шансів на

перемогу за кілька ходів до рішення про визнання поразки середній показник важливості складності ситуації становить 0.075. В той же час вплив часового обмеження дорівнює 0.0116, а досвіду - 0.0049.

В процесі дослідження двічі проведено перевірку важливості вхідних параметрів: на етапі аналізу даних визначено коефіцієнти кореляції та на етапі аналізу навчених моделей методом перестановки значень. Обидва способи показали такий порядок значущості: складність ситуації, часове обмеження, досвід. Проте виявлено суттєву відмінність у показниках: за коефіцієнтами кореляції часові обмеження на 36% менш важливі, ніж складність ситуації, а за втратою точності при перестановці значень - на 85%.

Висновки. Аналіз особливостей прийняття рішення людиною є складним завданням навіть при наявності великої кількості даних. При вивченні реальних ситуацій часто дослідники можуть опиратися лише на кілька параметрів: певний показник досвіду особи, обставини, в яких вона опинилася, та часові обмеження. Саме такі дані отримано з шахових партій, де гравці постійно стикаються з потребою приймати рішення. Одним з найважливіших рішень у цій грі є можливість визнання поразки - втрати контролю над ситуацією.

Завдяки аналізу сформованого набору даних виявлено, що прогнозувати поведінку більш досвідчених у сфері людей простіше, оскільки більш помітними є закономірності у прийнятих ними рішеннях. Крім цього професіонали здатні не лише довше не допускати помилок, але й швидше розуміти та визнавати втрату контролю над ситуацією.

Проведене дослідження показало, що отримані дані дозволяють прогнозувати рішення такого типу. У процесі навчання моделей нейронної мережі, випадкового лісу та градієнтного бустингу XGBoost вдалося отримати точність передбачення більше 90%. Що, в свою чергу, означає можливість правильно визначати, чи визнає гравець безнадійність ситуації, у переважній більшості випадків.

Проте прогнозування рішення людини у цьому випадку є лише бінарною класифікацією із задовільною точністю, а остаточна мета створення моделей полягає у аналізі їх роботи для порівняння рівня впливу параметрів. Завдяки визначенню залежностей у даних через коефіцієнти кореляції Спірмена, а також співставлення цих значень з важливістю наявних факторів на роботу моделей, виявлено суттєвий вплив складності ситуації на бажання людини визнати втрату контролю над процесами. Роль часового обмеження теж доволі відчутна, на відміну від рівня навичок гравців.

Список літератури

1. N. R. Hakobyan, I. Sotnyk, Y. Chortok, and A. Khachatryan, "On the issue of rationality of economic choice in the context of the psychological processes of anomie," “Katchar” Collection of Scientific Articles International Scientific-Educational Center NAS RA, pp. 1–13, 2024. [Online]. Available: https://doi.org/10.54503/2579-2903-2024.1-13

2. I. Molina, E. Molina-Perez, F. Sobrino, M. A. Tellez-Rojas, H. C. Zamora-Maldonado, M. Plaza-Ferreira, Y. Orozco, V. Espinoza-Juarez, L. Serra-Barragán, and A. De Unanue, "Current research trends on cognition, integrative complexity, and decision-making: A systematic literature review using activity theory and neuroscience," Frontiers in Psychology, vol. 14, Art. no. 1156696, 2023. [Online]. Available: https://doi.org/10.3389/fpsyg.2023.1156696

3. Lichess, "Lichess open database." [Online]. Available: https://database.lichess.org

4. S. Tang, Y. Wang, and C. Jin, "Is Elo rating reliable? A study under model misspecification," arXiv, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2502.10985

5. Stockfish, "Stockfish chess engine." [Online]. Available: https://stockfishchess.org

Ілечко Р.І.(1), Ілечко В.І.(2)
аспірант 3 курсу ІКНІ НУ ЛП, магістрантка 1 курсу ФПМІ ЛНУ ім. І. Франка
Науковий керівник к.т.н. доц. Цимбал Ю.В., кафедра АСУ НУ ЛП, к.т.н. доц. Мельничин А.В., кафедра теорії оптимальних процесів ЛНУ ім. І. Франка

ВПЛИВ РОЗМІРУ ЕКСТРАКТОРА ОЗНАК НА ТОЧНІСТЬ ВІЗУАЛЬНОЇ ГЕОЛОКАЛІЗАЦІЇ З БПЛА

Вступ. Сучасні методи візуальної геолокалізації вимагають високої точності та швидкості обробки даних при обмежених обчислювальних ресурсах. Це особливо стосується методів, які використовують БПЛА. Sample4Geo [1] є одним із перспективних методів у цій галузі, який поєднує локальні та глобальні ознаки для підвищення точності геолокаційних припущень на основі зображень. Його ефективність значною мірою залежить від характеристик екстрактора ознак, зокрема розміру та архітектури нейронного екстрактору. Розгортання моделей на енергообмежених пристроях, таких як дрони, вимагає оптимального балансу між складністю каркасної нейронної мережі та якістю геолокаційних прогнозів. Незважаючи на те, що потужні моделі, такі як ConvNeXt Base [2], успішно використовуються для задач геолокалізації, питання щодо впливу масштабування екстрактору на загальну продуктивність системи залишається відкритим. У цьому контексті аналіз впливу масштабів каркасної нейромережі на точність роботи Sample4Geo [1] є актуальним і практично значущим для створення компактних, але високоточних моделей візуальної геолокалізації БПЛА.

Постановка задачі. У цьому дослідженні поставлено задачу — визначити, якою мірою розмір екстрактора ознак (каркасної нейромережі) впливає на якість роботи моделі візуальної геолокалізації на базі Sample4Geo [1]. Передбачається проведення порівняльного аналізу перформансу моделей з різними конфігураціями нейронного екстрактору з метою виявлення оптимального співвідношення між точністю геолокації та складністю моделі. Об’єктом дослідження виступає система візуальної геолокалізації на основі глибоких нейронних мереж з використанням БПЛА. Предметом дослідження є вплив розміру та архітектури екстрактора ознак на точність роботи моделі Sample4Geo [1]. Основна мета дослідження полягає в тому, щоб визначити оптимальний розмір екстрактора ознак, що забезпечить точну локацію та прийнятні обчислювальні витрати при використанні моделі Sample4Geo [1] для задач візуальної геолокалізації з БПЛА.

Основний матеріал. У результаті проведеного дослідження було обґрунтовано взаємозв’язок між розміром каркасної нейронної мережі (екстрактора ознак) та якістю візуальної геолокалізації в межах архітектури Sample4Geo [1]. Аналіз показав, що збільшення глибини та кількості параметрів екстрактора, зокрема при використанні ConvNeXt Base [2], дійсно сприяє покращенню точності моделі за рахунок якіснішого представлення вхідних зображень у дескрипторному просторі. Більш складні моделі мають здатність вловлювати тонші відмінності у візуальних ознаках, що особливо важливо при локалізації в умовах слабкої структурної виразності місцевості. Проте водночас зростає обчислювальна складність та енергоспоживання, що критично для застосування на борту БПЛА з обмеженими апаратними ресурсами. Цей компроміс між точністю і ресурсомісткістю є ключовим у проектуванні систем автономної навігації.

У межах експериментальної частини було проведено порівняльний аналіз ефективності різних нейронних екстракторів у задачі візуальної геолокалізації. Основна мета — дослідити, як розмір та складність екстрактора ознак впливають на точність геолокалізації, а також визначити оптимальний баланс між якістю локалізації та обчислювальними витратами. Найвищі показники точності (Recall@1 = 92.37%, AP = 93.55%) були досягнуті при використанні потужного скелету ConvNeXt-Base [2], що має понад 88 млн. тренувальних параметрів і розмір моделі 354 МБ. Однак така точність супроводжується значними обчислювальними затратами, що робить дану конфігурацію малопридатною для використання на бортових системах з обмеженим енергобюджетом і продуктивністю.

Цікаво, що модель ConvNeXt-Nano [2], яка має у шість разів менше параметрів (15.6 млн.) та менший розмір (62.4 МБ), демонструє лише незначне погіршення точності (Recall@1 = 85.12%, AP = 87.48%). Це свідчить про її високу ефективність та потенціал до практичного застосування. Подібна ситуація спостерігається і з моделлю ViT-Tiny [3], яка, попри ще менший розмір (22.9 МБ) і кількість параметрів (5.7 млн.), забезпечує Recall@1 на рівні 78.27% та AP = 81.33%, перевершуючи більшу за параметрами модель MobileNetV3 [4], яка вважалась оптимізованою для мобільних застосувань. Такий результат підкреслює важливість не лише кількості параметрів, але й архітектурних особливостей моделі.

Водночас, моделі з найменшою кількістю параметрів, зокрема LeViT-128s [5], показали критично низькі результати (Recall@1 = 14.27%, AP = 18.74%), що вказує на недостатню потужність такого екстрактора для даного завдання.

Окрему увагу привертає порівняння моделей ViT-Small [3] та ViT-Tiny [3]. Хоча друга є спрощеною версією першої з меншим обсягом параметрів (5.7 млн. проти 22.9 млн.), вона демонструє кращі результати (Recall@1 = 78.27% проти 71.97%). Це може свідчити про те, що зменшення розміру моделі у випадку ViT [3] не завжди призводить до деградації якості. Можливо, ViT-Tiny [3] краще узгоджується з архітектурою Sample4Geo [1] або є менш схильною до перенавчання на обмежених даних, тоді як більша ViT-Small [3] могла втратити здатність до узагальнення через перенавантаження параметрів. Така поведінка свідчить про необхідність тонкого налаштування моделей до конкретного застосування замість сліпого масштабування.

Отримані результати демонструють чітку залежність між розміром екстрактора ознак і точністю геолокалізації. Проте, спостерігається також ефект насичення: збільшення розміру моделі понад певний поріг дає все менший приріст точності при значному зростанні обчислювальних витрат. Це дозволяє сформулювати думку про наявність зони компромісу між складністю нейромережевої моделі та її ефективністю в контексті геолокаційних задач. Таким чином, ConvNeXt-Nano [2] та ViT-Tiny [3] можна розглядати як оптимальні рішення, що забезпечують прийнятну точність при суттєвій економії ресурсів. Вони становлять практичну цінність для реалізації onboard-систем геолокації на БПЛА, де кожен мегабайт і міліват енергії мають значення.

Висновки. У результаті дослідження встановлено, що зменшення розміру екстрактора ознак у моделі Sample4Geo [1] може суттєво знизити обчислювальні витрати при незначній втраті точності, що дозволяє ефективно використовувати візуальну геолокалізацію на БПЛА з обмеженими ресурсами. Наукова новизна полягає в емпіричному визначенні оптимального співвідношення між розміром моделі та якістю локалізації, а практичне значення — у можливості адаптації методу для застосування в реальних умовах автономної навігації.

Список літератури

1. Deuser, F., Habel, K., & Oswald, N. (10 2023). Sample4Geo: Hard Negative Sampling For Cross-View Geo-Localisation. 16801–16810. doi:10.1109/ICCV51070.2023.01545
2. Todi, A., Narula, N., Sharma, M., & Gupta, U. (2023). ConvNext: A Contemporary Architecture for Convolutional Neural Networks for Image Classification. *2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, 1–6. doi:10.1109/CISCT57197.2023.10351320
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ArXiv, abs/2010.11929. Retrieved from https://api.semanticscholar.org/CorpusID:225039882
4. Qian, S., Ning, C., & Hu, Y. (2021). MobileNetV3 for Image Classification. 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 490–497. doi:10.1109/ICBAIE52039.2021.9389905
5. Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., & Douze, M. (2021). LeViT: a Vision Transformer in ConvNet’s Clothing for Faster Inference. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2104.01136

Лучит Н.В.
аспірант 1 курсу ІКНІ НУ «Львівська Політехніка»
Науковий керівник д.т.н. Медиковський М.О., ІКНІ НУ «Львівська Політехніка»

ЗАСТОСУВАННЯ МАРКОВСЬКИХ ПРОЦЕСІВ У ЗАДАЧАХ БАГАТОКРИТЕРІАЛЬНОЇ НЕЛІНІЙНОЇ ОПТИМІЗАЦІЇ

Вступ. У сучасному світі дедалі більше прикладних задач характеризуються складністю моделювання через зростання кількості зовнішніх впливів, взаємозв'язків і обмежень, що часто є нелінійними. Прикладом таких задач може бути оптимізація енергоменеджменту на підприємствах або в мікромережах, управління роботами та маніпуляторами, формування інвестиційних портфелів тощо. Зазвичай ці задачі формалізуються як багаторівнева цільова функція з множиною нелінійних обмежень. Щоб уникнути складного процесу лінеаризації, задачі такого типу доцільно моделювати у вигляді марковського процесу прийняття рішень (MDP – Markov Decision Process), що дозволяє застосовувати сучасні підходи глибинного навчання з підкріпленням [1].

Постановка задачі. Об'єкт дослідження – процес розв'язання задачі багатокритеріальної оптимізації, що містить нелінійні обмеження та складні зв'язки між змінними. Предмет дослідження – методи навчання з підкріпленням Proximal Policy Optimization (PPO) для пошуку розв'язку задачі оптимізації, поданої у вигляді марковського процесу прийняття рішень (MDP). Мета дослідження – визначити ефективність та особливості застосування PPO для розв'язання задачі багатокритеріальної оптимізації в умовах нелінійності та невизначеності середовища.

Основний матеріал. У цьому дослідженні розглянуто спосіб розв'язання задачі багатокритеріальної оптимізації за допомогою алгоритму глибинного навчання з підкріпленням PPO. Алгоритм PPO розроблений у 2017 р. Джоном Шульманом (співзасновник компанії OpenAI) та сьогодні є одним з найпопулярніших засобів для реалізації методу навчання з підкріпленням [2]. Цей алгоритм є покращеною версією вже існуючого алгоритму TRPO (Trust Region Policy Optimization). Процес навчання та оптимізації моделі агента копіює підхід до виховання дітей, шляхом здобуття досвіду та оцінки за допомогою системи штрафів та нагород, у вигляді зміни політики поведінки агента у відповідному напрямку.

Алгоритм PPO зазвичай реалізується за допомогою архітектури актор-критик (actor-critic) яка передбачає використання двох нейронних мереж – актора та критика. Актор (policy network) – отримує кортеж, що описує стан середовища та обирає наступну дію. Критик (value network) – оцінює очікувану сумарну винагороду поточного стану. Процес навчання цих моделей включає декілька етапів:

1. Зібрання траєкторій (послідовностей кортежів, що складаються зі стану, дії, нагороди та наступного стану).
2. Обчислення переваг (advantages) та повернень (returns), тобто сумарних винагород.
3. Розрахунок функції втрат актора (policy loss) та функції втрат критика (value loss).
4. Оновлення параметрів нейромереж з метою мінімізації цих функцій втрат.

У базовій реалізації політики градієнтного спуску (basic policy gradient method) функція втрат визначається залежністю $L^{PG}(\theta) = \mathbb{E}_t[\log \pi_\theta(a_t|s_t) \cdot A_t]$, де $\pi_\theta(a_t|s_t)$ - імовірність дії (a_t), яку актор (політика) передбачив в стані (s_t), $A_t = R_t - V(s_t)$ – перевага, це оцінка наскільки ця дія була кращою за очікувану винагороду. Такий підхід реалізації функції втрат має один суттєвий недолік – оновлення параметрів нейромереж стає надто чутливим до різких змін в політиці, адже значення переваги безпосередньо впливає на крок зміни політики в позитивну чи негативну сторону [3]. Алгоритм PPO передбачає зміну в підході до обчислення функції втрати шляхом обрахунку сурогатної функції втрат з обмеженням (clipped surrogate loss) $L^{CLIP}(\theta) = \widehat{\mathbb{E}}_t\left[\min\left(r_t(\theta) \cdot \hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \cdot \hat{A}_t\right)\right]$, де $r_t(\theta)$ – це коефіцієнт зміни політики (policy ratio), що визначається залежністю $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{old}}(a_t \mid s_t)$, як відношення нової політики $\pi_\theta(a_t \mid s_t)$ до попередньої $\pi_{\theta_{old}}(a_t \mid s_t)$, функція $clip()$ обмежує це відношення на інтервалі $[1-\epsilon, 1+\epsilon]$, де ϵ – гіперпараметр (в цьому дослідженні $\epsilon = 0.2$). Таким чином

алгоритм PPO вирішує основний недолік звичайного алгоритму policy based, що робить його стійким до різких змін в політиці шляхом встановлення граничних значень зміни політики за допомогою гіперпараметру епсилон.

Моделювання багатокритеріальної задачі оптимізації не залежить від реалізації алгоритму PPO, завдяки чому використання такого підходу є гнучким та універсальним. Взаємодія між математичною моделлю середовища та агентом відбувається за принципом «чорної скриньки». Середовище лише повертає стан (зазвичай це масив значень, що чітко описують ситуацію в якій перебуває агент, до прикладу: координати положення агента, рівень навантаження силових установок, рівень генерації різних видів енергії тощо), виконує дію (переводить згідно з отриманою командою (action) середовище з поточного стану в наступний), обчислює та надає нагороду за виконану дію. Таку модель взаємодії називають марковським процесом прийняття рішень (MDP) [4], відтак застосування методів глибинного навчання з підкріпленням зводиться до вирішення двох задач: технічної реалізації агента (імплементація архітектури актор-критик та обчислення сурогатної функції втрати), а також моделювання середовища. Ці задачі можуть бути виконані незалежно, а коригування агента, зазвичай, відбувається за допомогою зміни зовнішніх гіперпараметрів, що дозволяють більш точно налаштувати процес навчання відповідно до моделі середовища. Крім того, важливим аспектом є формування ефективної системи нагороди та штрафів, яка безпосередньо впливає на швидкість і якість навчання агента. У багатьох випадках зміна середовища або додавання нових змінних не вимагає повного перенавчання агента, а лише коригування його гіперпараметрів. Також, архітектура актор-критик дозволяє швидко масштабувати середовище під задачі з великою кількістю станів і дій [5]. Це робить алгоритм PPO придатним для застосування у складних динамічних системах, де класичні методи оптимізації виявляються малоефективними.

Висновки. У ході дослідження підтверджено можливості застосування алгоритму глибинного навчання з підкріпленням PPO для розв'язання задачі багатокритеріальної оптимізації. Показано, що цей підхід дозволяє ефективно працювати з нелінійними обмеженнями та складними середовищами без необхідності лінеаризації моделі. Особливістю PPO є стабільність процесу навчання завдяки застосуванню механізму обмеження змін у політиці агента (clipped surrogate loss). При тестовій реалізації визначено потенціал використання PPO для широкого класу задач оптимізації в інженерних та технічних системах. Перспективами подальших досліджень є масштабування підходу на складніші середовища та покращення ефективності за допомогою адаптивного налаштування гіперпараметрів.

Список літератури

1. Zhou S., Hu Z., Gu W., Jiang M., Chen M., Hong Q., Booth C. Combined heat and power system intelligent economic dispatch: A deep reinforcement learning approach // International Journal of Electrical Power & Energy Systems. – 2020. – Vol. 120. – Article ID: 106016. – ISSN 0142-0615. – DOI: https://doi.org/10.1016/j.ijepes.2020.106016.

2. Hasani Zadeh B.M., Mansouri N. PPO: a new nature-inspired metaheuristic algorithm based on predation for optimization // Soft Comput. – 2022. – Vol. 26. – P. 1331–1402. – DOI: https://doi.org/10.1007/s00500-021-06404-x.

3. Del Rio A., Jimenez D., Serrano J. Comparative Analysis of A3C and PPO Algorithms in Reinforcement Learning: A Survey on General Environments // IEEE Access. – 2024. – Vol. 12. – P. 12345–12360. – DOI: https://doi.org/10.1109/access.2024.3472473.

4. Linrong W., Xiang F., Ruifen Z., Zhengran H., Guilan W., Haixiao Z. Energy management of integrated energy system in the park under multiple time scales // AIMS Energy – 2024. – Vol. 12, No. 3 – P. 639-663. – DOI: https://doi.org/10.3934/energy.2024030

5. Singh A., Panigrahi B.K. Prosumer Cost Efficiency and Ensuring Grid Stability Through a Hierarchical PPO Framework in Decentralized Community Energy Management // IEEE Access – 2025. – Vol. 13. - P. 38368-38386. – DOI: https://doi.org/10.1109/access.2025.3542873.

Мамчур Т. Б.
аспірант 2 курсу ІКНІ НУ ЛП
Науковий керівник к.т.н. Опотяк Ю.В., кафедра АСУ НУ ЛП

ВДОСКОНАЛЕННЯ МЕТОДІВ НЕЙРОМЕРЕЖЕВОГО КРИПТОГРАФІЧНОГО ЗАХИСТУ ДАНИХ В РЕАЛЬНОМУ ЧАСІ

Вступ. Для мобільних робототехнічних платформ (МРП) важливою проблемою є забезпечення криптографічного захисту передачі даних. Вирішення такої проблеми потребує розроблення методів нейромережевого криптографічного захисту даних, які орієнтовані на використання у бортових системах. Під час розроблення бортових систем криптографічного захисту даних необхідно забезпечити режим реального часу, підвищити криптостійкість та зменшити масу, габарити, енергоспоживання та вартість. Одним із способів забезпечення таких вимог є використання автоасоціативної нейромережі прямого поширення, яка навчається на підставі методу головних компонент. Особливістю таких нейромереж є можливість наперед обчислити вагові коефіцієнти, використати таблично-алгоритмічний метод для реалізації нейроподібних елементів з використанням базису елементарних арифметичних операцій. Для нейромережевого шифрування та дешифрування даних пропонують використати симетричні ключі, які містять коди маскування, архітектуру нейромережі та матриці вагових коефіцієнтів.

Постановка задачі. Об'єкт дослідження – розроблення методів криптографічного захисту даних.

Предмет дослідження – методи та засоби розроблення системи нейромережевого криптографічного захисту даних у реальному часі, що базуються на таблично-алгоритмічному обчисленні скалярного добутку векторів з урахуванням впливу вхідних параметрів і забезпечують підвищену швидкодію та криптографічну стійкість системи.

Мета роботи – розробити метод нейромережевого криптографічного захисту даних у реальному часі, що забезпечує підвищену швидкодію та криптографічну стійкість.

Основний матеріал. Високих техніко-економічних показників бортових систем криптографічного захисту даних досягають способом використання проблемно орієнтованого підходу, сучасної елементної бази (мікроконтролерів, програмованих логічних інтегральних схем (ПЛІС) типу FPGA), розроблення нових методів, алгоритмів і структур. Реалізація бортових систем криптографічного захисту на підставі проблемно-орієнтованого підходу передбачає використання універсального процесорного ядра, доповненого спеціалізованими апаратними засобами, які реалізують нейроподібні елементи. Поєднання універсальних і спеціалізованих, програмних та апаратних засобів забезпечить створення малогабаритних систем нейромережевого криптографічного захисту даних у реальному часі зі швидкою зміною ключів шифрування та дешифрування даних.

Нейромережевого криптографічного шифрування та дешифрування даних у реальному часі досягають унаслідок використання розпаралелення процесів шифрування та дешифрування даних, апаратної реалізації нейроподібних елементів на базі багатоопérандного підходу і макрочасткових таблиць.

Для досягнення зазначеної мети виконано наступні завдання дослідження.

Розроблено метод криптографічного захисту даних із використанням нейромереж прямого поширення автоасоціативного типу на підставі парадигми послідовних геометричних перетворень, застосування якого забезпечує ефективну таблично-алгоритмічну реалізацію нейроподібних елементів.

Удосконалено таблично-алгоритмічний метод обчислення скалярного добутку векторів способом використання вагових коефіцієнтів у форматі з плавучою комою, що дало змогу підвищити точність обчислень, забезпечити гнучкість адаптації до різнорідних типів вхідних даних та зменшити кількість помилок заокруглення під час оброблення їхніх великих масивів з реальними значеннями.

Розроблено базову структуру таблично-алгоритмічного пристрою обчислення скалярного добутку векторів, використання якого забезпечує зменшення тривалості його розроблення зі заданими параметрами.

Розроблено структуру проблемно орієнтованої системи нейромережевого криптографічного захисту даних, яка забезпечує можливість вибору архітектури нейромережі, зміни маски та таблиць макрочасткових добутків елементів векторів.

Проєктування спеціалізованих апаратних засобів підсистеми нейромережевого шифрування даних виконували мовою програмування апаратури VHDL у середовищі розроблення Quartus II вер. 13.1 з використанням бібліотек середовища розроблення. Середовище розроблення Quartus II підтримує весь процес проєктування спеціалізованих апаратних засобів, починаючи із введення проєкту користувачем і завершуючи програмуванням ПЛІС, налагодженням як самої мікросхеми, так і засобів загалом. Для реалізації спеціалізованих апаратних засобів нейромережевого шифрування даних на базі FPGA EP3C16F484C6 сімейства Cyclone III п 3053 логічних елементів із 15408 логічних елементів і 745 регістрів. Час, потрібний для шифрування одного вектора вхідних даних, становить приблизно 160 наносекунд.

Отримані результати підтверджують ефективність запропонованого методу нейромережевого шифрування даних, реалізованого на базі FPGA. Використання лише 3053 логічних елементів (менше 20 % від доступних ресурсів кристала) та 745 регістрів свідчить про раціональне використання апаратної бази, що є критично важливим для бортових систем з обмеженими ресурсами. Швидкодія на рівні 160 наносекунд для оброблення одного вектора вхідних даних демонструє доцільність застосування таблично-алгоритмічного підходу до реалізації нейроподібних елементів, що забезпечує виконання криптографічних операцій у режимі реального часу.

Висновки. Розроблено проблемно орієнтовану систему нейромережевого криптографічного захисту даних у реальному часі, що забезпечує підвищену її швидкодію та криптографічну стійкість.

Розроблено метод нейромережевого криптографічного захисту даних, який за рахунок використання нейромереж прямого поширення автоасоціативного типу на підставі парадигми моделі послідовних геометричних перетворень, попереднього обчислення вагових коефіцієнтів та таблично-алгоритмічної реалізації нейроподібних елементів забезпечує високу її криптографічну стійкість, шифрування-дешифрування даних у реальному часі та апаратно-програмну реалізацію з високими техніко-експлуатаційними характеристиками.

Виконано моделювання спеціалізованих апаратних засобів нейромережевого шифрування даних мовою програмування апаратури VHDL у середовищі розроблення Quartus II вер. 13.1 з використанням FPGA EP3C16F484C6 сімейства Cyclone III та визначено, шо тривалість шифрування одного вектора вхідних даних становить приблизно 160 нс, а для їх реалізації потрібно 3053 логічних елементів і 745 регістрів.

Список літератури

1. Awad, A. I., Furnell, S., Paprzycki, M., & Sharma, S. K. (Eds.). (2021). Security in Cyber-Physical Systems: Foundations and Applications. Springer, 105–112, https://doi.org/10.1007/978-3-030-67361-1

2. Mahmoud, M. M., Gad, R. M., Younis, M., & Alghamdi, T. A. (2021). Secure Communication Protocols for Mobile Robotic Networks: Challenges and Solutions. IEEE Communications Surveys & Tutorials, 23(2), 1287–1312. https://doi.org/10.1109/COMST.2020.3048767

3. Delacour, K. J., & Wolf, C. (2020). A Survey of Neural Network Applications to Cryptography. Neural Computing and Applications, 32(17), 13349–13374. https://doi.org/10.1007/s00521-020-04859-9

4. Tsmots, I., Teslyuk, V., Łukaszewicz, A., Lukashchuk, Y., Kazymyra, I., Holovatyy, A., & Opotyak, Y. (2023). An approach to the implementation of a neural network for cryptographic protection of data transmission at UAV. Drones, 7(8), article ID 507. https://doi.org/10.3390/drones7080507

Юр’єв Д.Л.
магістрант 5 курсу ІІБРТ
Науковий керівник к.е.н., доцент Тимошенко Л.М., кафедра КБПЗ,
Національний університет «Одеська політехніка»

СИСТЕМА МОНІТОРИНГУ КІБЕРСИТУАЦІЙНОЇ ОБІЗНАНОСТІ

Вступ. Термін «ситуаційна обізнаність» з'явився у військовій галузі ще під час Першої світової війни, але з розвитком інформаційних технологій він набуває подальшого розвитку та трансформації. Ситуаційна обізнаність означає здатність отримувати інформацію про поточний стан середовища (системи) та на основі знань формувати висновки про необхідні дії [1]. Головною перевагою в сучасній війні росії проти України вважають повну інформаційну обізнаність на полі бою, розуміння того, де знаходиться ворог, інформацію про його чисельність і боєздатність тощо. Україна представила НАТО систему ситуаційної обізнаності Delta, що забезпечує військових різними даними про противника та допомагає координувати сили на полі бою, та використовується для планування операцій та бойових завдань, координації з іншими підрозділами, захищеного обміну інформацією тощо [2].

Перехід війни в кіберпростір сприяє зростанню кількості практичних розробок і наукових праць з проблематики кіберситуаційної обізнаності, визначає актуальність досліджень з оцінювання та моніторингу кіберситуаційної обізнаності та інформаційної безпеки систем в цілому для різних сфер державного управління та держави в цілому.

Постановка задачі. Об'єктом дослідження є кіберситуаційна обізнаність співробітників об’єкту критичної інфраструктури. Предметом дослідження є оцінювання кіберситуаційної обізнаності співробітників Одеської обласної державної (військової) адміністрації. Головна мета даної роботи - дослідження систем кіберситуаційної обізнаності, інструментів для опису стану ситуаційної обізнаності, методів оцінювання кіберситуаційної обізнаності та розробка програмної системи тестування та моніторингу кіберситуаційної обізнаності.

Основний матеріал. Кіберситуаційна обізнаність (Cyber Situational Awareness) – область досліджень, пов'язана із застосуванням методів оцінювання захищеності інформаційних систем та штучного інтелекту в кібербезпеці, спрямована на підвищення обізнаності про можливі ситуації порушень кібербезпеки та автоматичне виявлення кіберзагроз. Згідно моделі Ендслі (Endsley) [3], стан ситуаційної обізнаності системи є результатом процесу аналізу та оцінки ситуації.

Для оцінювання кіберситуаційної обізнаності співробітників об’єкту критичної інфраструктури у цій роботі використано метод тестування. Виходячи з вимог та принципів комп’ютерного тестування, складено тестові завдання з трьохот питань, на кожне з яких запропоновано три варіанти відповіді. Для визначення експертних оцінок, що є вагами кожної відповіді, використано метод аналізу ієрархій (MAI) Томаса Сааті [4].

Ієрархічну модель системи розглянемо як цілеспрямовану інформаційно-управлінську структуру з обов’язковим урахуванням ієрархічних рівнів її організації (рисунок 1).

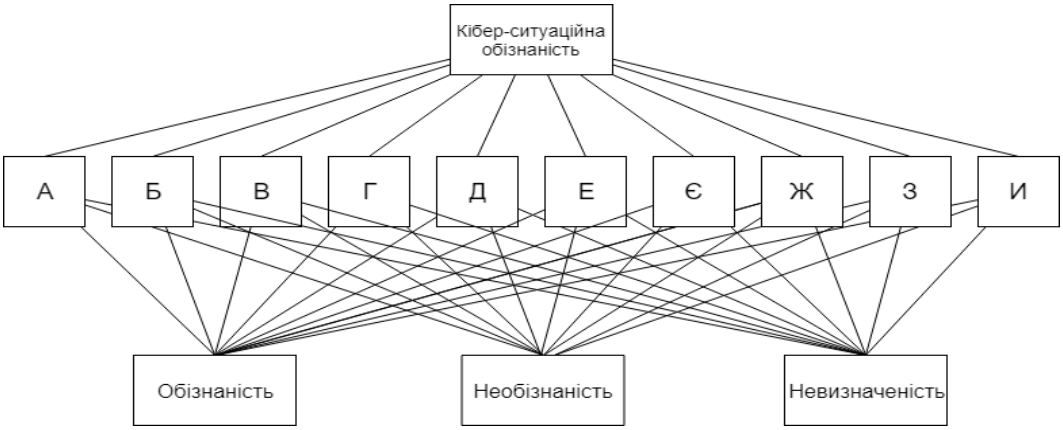


Рисунок 1 – Ієрархічна модель згідно з методом Сааті

На першому рівні (ціль) системи – кіберситуаційна обізнаність, на другому (критерії) – сфери кіберзахисту, третьому (альтернативи) – рівні обізнаності. У якості критеріїв А-И обрано для прикладу різні сфери кіберзахисту: фізичний захист; парольний; антивірусний; організаційний; програмний; від соціальної інженерії; при роботі в Інтернет; зберігання, видалення даних; використання мобільних пристроїв; робота з конфіденційними документами.

Відповідно до МАІ складено матрицю порівняння критеріїв, матриці порівняння альтернатив за кожним критерієм, матриці ваги альтернатив і ваги критеріїв. В результаті дослідження з'ясована експертна оцінка кіберситуаційної обізнаності співробітників. Для перевірки узгодженості експертних оцінок знайдено λ_{max} – найбільше власне число матриці парних порівнянь, індекс узгодженості (I_y) та відношення узгодженості (B_y). Узгодження експертних оцінок кіберситуаційної обізнаності співробітників зображено на рисунку 2.

Фізичний захист	Обізнаність	Необізнаність	Невизначеність
Обізнаність	1	7	5
Необізнаність	1/7	1	1/3
Невизначеність	1/5	3	1

$\lambda_{max} = 3,06.$

$I_y = 0,03$

$B_y = 0,052$ – матриця узгоджена.

Рисунок 2 – Узгодження експертних оцінок

Враховуючи одержані ваги відповідей у тестах, використовуючи шкалу Лайкерта рівнів обізнаності, можемо розрахувати рівень кібер-ситуаційної обізнаності. Для реалізації запропонованого оцінювання кібер-ситуаційної обізнаності використано Python та Michrosoft Excel. Виконано експериментальне дослідження тестування ситуаційної обізнаності та аналіз і моніторинг одержаних результатів. Розроблені заходи підвищення рівня кіберситуаційної обізнаності та рекомендації по їх впровадженню для покращення захищеності інформаційних ресурсів Одеської обласної державної (військової) державної адміністрації.

Висновки. Проведені дослідження дозволяють зробити висновок, що в якості інструменту опису стану ситуаційної обізнаності доцільно використовувати модель Ендслі, методи оцінювання ситуаційної обізнаності – комп'ютерне тестування та метод аналізу ієрархій – дозволили реалізувати програмну систему моніторингу кіберситуаційної обізнаності. Очевидно, що забезпечення ситуаційної обізнаності є вкрай важливим сьогодні для управління безпекою об'єктів критичної інфраструктури та відбиття постійних інформаційних атак ворога.

Список літератури

1. Буров Є.В., Микіч Х.І. Формальна модель опрацювання знань у системах із ситуаційною обізнаністю *Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі*. 2017. № 872. С. 25-35.

2. Українську систему ситуаційної обізнаності презентували на щорічному заході НАТО [Електронний ресурс].-Режим доступу: https://censor.net/ua/n3376484

3. Тимошенко Л.М., Єрьоменко А.І. Використання моделі Ендслі для розробки систем кібер-ситуаційної обізнаності. *Матеріали XIX Всеукраїнської науково-технічної конференції «Стан, досягнення і перспективи інформаційних систем і технологій»*. Одеса: ОНАХТ, 2020. С. 74-75.

4. Saaty Thomas L. Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process (1994). Pittsburgh: RWS. ISBN 0-9620317-6-3.

Шевчук Ю.А.
студент 4 курсу ІІБРТ, Національний університет «Одеська політехніка»
Науковий керівник к.т.н., доцент Агаджанян О.В., кафедра КБПЗ
Національний університет «Одеська політехніка»

РОЗРОБКА ЗАХИЩЕНОГО ЧАТ-БОТУ «УНІВЕРСИТЕТСЬКИЙ ПОМІЧНИК ПЕРШОКУРСНИКУ»

Вступ. Цифрова трансформація сучасної вищої освіти створює передумови для впровадження нових форм комунікації між студентами та адміністрацією закладів. Зокрема, першокурсники як нові учасники освітнього процесу часто відчувають труднощі з орієнтуванням у навчальному середовищі, пошуком необхідної інформації, розумінням процедур та розкладу занять. Це знижує їхню адаптацію, викликає емоційне напруження і, зрештою, може негативно вплинути на мотивацію до навчання.

Ситуація ускладнюється ще й тим, що велика кількість запитів, які виникають у студентів, є повторюваними, типовими та адміністративно обумовленими. Працівники деканатів або інформаційних служб змушені постійно відповідати на однакові звернення, що не є ефективним використанням ресурсів. У той же час студенти очікують швидкої реакції, особливо у випадках, пов'язаних із дедлайнами, розкладом або змінами в освітньому процесі.

В умовах широкого використання месенджерів, зокрема серед молоді, виникає природна потреба в інтерактивному цифровому помічнику, який міг би здійснювати швидке надання довідкової інформації, допомагати у навігації та знижувати навантаження на персонал. Одним із найефективніших інструментів для цього є чат-бот — програмна система, що працює у середовищі обміну повідомленнями, забезпечуючи природну та зрозумілу взаємодію з користувачем.

Однак більшість наявних чат-ботів або не забезпечують належного рівня захисту персональних даних, або мають обмежений функціонал. У таких умовах особливо актуальною є розробка чат-бота з локальним розгортанням, надійними механізмами безпеки, адаптованого до освітніх потреб українських університетів.

Постановка задачі. Об'єкт дослідження – інформаційна система, що забезпечує комунікацію між студентами та університетськими сервісами. Предмет дослідження – процеси проєктування та реалізації чат-бота з урахуванням вимог інформаційної безпеки, масштабованості й зручності використання. Мета дослідження – розробити безпечного, функціонального чат-бота для першокурсників, який включає в себе інструменти реєстрації, доступу до довідкової інформації, навчального розкладу, контактів факультетів та інтерактивного спілкування англійською мовою за допомогою вбудованої мовної моделі.

Основний матеріал. Розроблений чат-бот реалізовано на мові програмування Python із застосуванням асинхронного фреймворку, що дозволяє ефективно працювати з великою кількістю одночасних запитів [1]. Обробка діалогів реалізована через механізм скінченної автомати, який дозволяє структурувати сценарії спілкування, зокрема під час проходження користувачем етапу реєстрації. Для тимчасового зберігання станів та обмеження частоти запитів використовується система Redis, яка функціонує в оперативній пам'яті, забезпечуючи високу швидкість доступу до даних [2].

Збереження персональної інформації реалізується через реляційну базу даних PostgreSQL. Це дозволяє не лише зберігати структуру довідкових розділів, а й підтримувати фільтрацію, сортування, пошук та інші функції, необхідні для масштабованого використання. Архітектура чат-бота побудована за модульним принципом, де кожен компонент ізольовано в межах окремого середовища виконання. Усі модулі, зокрема база даних, система зберігання станів, інтерфейс взаємодії, а також компонент генерації тексту, розгортаються за допомогою Docker та управляються через Docker Compose.

Однією з ключових інновацій проєкту є можливість інтерактивного спілкування англійською мовою через вбудовану локальну мовну модель [3-4]. Для цього використано Ollama — програмну платформу з відкритим кодом, що дозволяє розгортати великі мовні моделі

без підключення до зовнішніх API [5]. Впроваджена модель працює автономно, генеруючи відповіді на запитання англійською та забезпечуючи повну конфіденційність даних. Інтеграція LLM-модуля у загальну архітектуру здійснюється за допомогою HTTP-запитів, які передають сформульоване питання до локального API і приймають відповідь у режимі потокової передачі.

Функціонал чат-бота включає реєстрацію нового користувача з перевіркою в базі даних, надання інформації про розклад занять, контакти факультетів, відповіді на поширені запитання з підтримкою пагінації, а також практика знань англійської мови. Навігація реалізована за допомогою кнопок і callback-запитів, що дозволяє зберегти простоту інтерфейсу.

Особливу увагу приділено інформаційній безпеці. Реалізовано механізм фільтрації вхідного контенту, який блокує повідомлення з підозрілими посиланнями або тегами масового згадування. Користувачі, що перевищують допустиму кількість запитів за одиницю часу, автоматично блокуються на заданий інтервал. Для забезпечення конфіденційності введено маскування телефонних номерів у логах. Логування подій здійснюється з урахуванням вимог до збереження критичної інформації для аудиту, при цьому без розкриття персональних даних.

У межах тестування було змодельовано декілька сценаріїв, зокрема надсилання заборонених повідомлень, перевищення ліміту запитів, взаємодія з поширеними запитаннями, перегляд розкладу та діалог з мовною моделлю. Усі перевірки засвідчили стабільну роботу системи та належне реагування на нестандартні ситуації.

Висновки. Розроблений чат-бот є прикладом ефективного застосування сучасних технологій у сфері цифрової підтримки студентів. Його модульна архітектура, локальна мовна модель та реалізовані механізми захисту роблять систему придатною для впровадження у будь-якому навчальному закладі. Новизна проєкту полягає у поєднанні локального виконання великої мовної моделі, засобів автоматичного контролю вхідного контенту, ізоляції сервісів та високого рівня інтегрованості між компонентами.

Практичне значення розробки полягає в її масштабованості, конфігурованості та відповідності вимогам захисту персональних даних. Запропоноване рішення може бути адаптоване до будь-якої навчальної інституції без залежності від хмарних сервісів, що особливо актуально в умовах обмеженого доступу до мережі або потреби у внутрішньому захищеному середовищі.

Подальший розвиток системи може включати розширення функціоналу, інтеграцію з системами управління навчанням, підтримку багатомовності, підключення аналітики або впровадження механізмів зворотного зв’язку. Отримані результати свідчать про доцільність використання подібних технологій у сфері освіти для підвищення ефективності взаємодії студентів із навчальними структурами.

Список літератури

1. Aiogram Documentation [Електронний ресурс]. URL: https://docs.aiogram.dev/ (дата звернення: 10.05.2025).

2. Redis Documentation [Електронний ресурс]. URL: https://redis.io/ (дата звернення: 11.05.2025).

3. Кушнерьов О. С., Вадько К. С., Кузченко А. О. Чат-бот на основі нейромережі для надання інформаційної підтримки бізнесу з позиції кібербезпеки // Сучасний захист інформації. 2024. № 4(60). С. 131–140. DOI: 10.31673/2409-7292.2024.040014.

4. Терлецька Т. Використання чат-ботів на основі великих мовних моделей у науково-педагогічній діяльності викладачів // Open Educational E-environment of Modern University. 2024. № 16. С. 194–209. URL: https://www.researchgate.net/publication/380184525 (дата звернення: 11.05.2025).

5. Ollama Documentation [Електронний ресурс]. URL: https://ollama.com (дата звернення: 11.05.2025).

Виліткова А. А.
студентка 4 курсу кафедри АСУ, ІКНІ, Національний університет "Львівська політехніка"
Науковий керівник док.філософії, асистент Казарян А.Г., кафедра АСУ, ІКНІ,
Національний університет "Львівська політехніка"

АВТОМАТИЗОВАНА СИСТЕМА ПЕРСОНАЛІЗОВАНОГО ПІДБОРУ ОДЯГУ З ВИКОРИСТАННЯМ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Вступ. Сучасні користувачі стикаються з труднощами у виборі відповідного одягу залежно від погодних умов, події та стилістичних уподобань. Існуючі рішення зосереджені на класифікації одягу або рекомендаціях на основі фіксованих правил, проте вони не забезпечують належної адаптації до змінних умов, таких як погода чи особисті стилістичні вподобання користувача. Використання штучного інтелекту (ШІ), зокрема великих мовних моделей, відкриває нові можливості у створенні персоналізованих рекомендацій. Аналіз наукових джерел показує, що більшість існуючих рішень зосереджені на класифікації одягу на основі зображень [1, 2]. Інші підходи передбачають використання моделей для визначення стилю або кольорової гармонії [3, 4]. Проте, відсутність інтеграції з актуальними погодними умовами та можливістю адаптації до уподобань користувача залишається проблемою.

Постановка задачі. Мета дослідження – розробити веб-застосунок цифрового помічника у виборі одягу, який використовує можливості ШІ для класифікації, збереження та підбору одягу відповідно до погодних умов та стилістичних уподобань користувача. Об'єктом дослідження є процес автоматизованого вибору одягу. Предметом дослідження є методи використання штучного інтелекту для аналізу та класифікації одягу.

Основний матеріал. Розроблена система базується на використанні веб-технологій (HTML, CSS, JavaScript) та хмарного сервісу Firebase для зберігання даних гардеробу. Веб-застосунок містить наступні сторінки: додавання одягу, перегляд гардеробу та автоматичний підбір образу на основі погодних умов та стилю. Завантажене зображення аналізується великою мовною моделлю OpenAI, яка визначає тип, підтип, стиль та інші атрибути одягу. Отримані дані зберігаються у хмарі, забезпечуючи їх доступність та захист. Функція підбору одягу реалізована за допомогою алгоритму сумісності, що враховує стилістичні та погодні критерії. Користувач обирає бажаний стиль, після чого система автоматично визначає погодні умови за допомогою Open-Meteo API та формує список відповідних речей з гардеробу. Розрахунок сумісності між елементами одягу враховує коефіцієнти для типів одягу та кольорової гамми, що дозволяє створювати збалансовані та стильні образи. У таблиці 1 наведені приклади коефіцієнтів сумісності між типами одягу.

Таблиця 1 - Приклади коефіцієнтів сумісності між типами одягу

Верхній одяг	**Нижній одяг**	**Коефіцієнт**
Футболка	Джинси	8
Сорочка	Штани	9
Блузка	Спідниця	9
Светр	Штани	8
Кеди	Шорти	9
Черевики	Джинси	9

Система також враховує сумісність кольорів, використовуючи заздалегідь визначені коефіцієнти для пар кольорів. Наприклад, чорний і білий мають високу сумісність з коефіцієнтом 9. При формуванні комплекту одягу система обчислює загальну сумісність кольорів, враховуючи всі можливі комбінації кольорів між верхом, низом і взуттям.

Коефіцієнти кольорів визначають, наскільки добре кольори з різних елементів одягу доповнюють один одного. Поєднуючи значення сумісності різних пар кольорів, система генерує

загальний бал сумісності для всього комплекту. При обчисленні загального балу сумісності образу враховується сума коефіцієнтів сумісності для всіх можливих парних комбінацій кольорів між верхнім одягом, нижнім одягом і взуттям. Наприклад, якщо верхній одяг чорного кольору, низ — білого, а взуття — синього, то розраховується сумісність для пар: чорний-білий, чорний-синій, білий-синій.

Приклад коефіцієнтів сумісності між кольорами одягу наведено в Таблиці 2.

Таблиця 2 - Приклади коефіцієнтів сумісності між кольорами одягу

Колір 1	**Колір 2**	**Коефіцієнт**
Чорний	Білий	9
Чорний	Синій	7
Білий	Червоний	7
Синій	Зелений	6
Зелений	Коричневий	6

Для кожної комбінації обчислюється сумарний бал сумісності. Цей бал включає коефіцієнти кольорової сумісності між парами елементів (наприклад, між верхом і низом, між низом і взуттям), а також коефіцієнти сумісності типів одягу (наприклад, сорочка з джинсами, або пальто з кофтами). Якщо присутній верхній шар одягу, він також враховується в розрахунках через сумісність кольору та типу. Формула загального коефіцієнта сумісності між елементами гардеробу має наступний вигляд:

$$S = \Sigma(C_i j * K_i j),$$

де S – загальний бал сумісності; $C_i j$ – коефіцієнт сумісності кольорів між і-м та j-м елементами; $K_i j$ – коефіцієнт сумісності типів одягу.

Після обчислення всіх можливих комбінацій система сортує їх за спаданням сумісності та повертає п’ять найкращих варіантів. Представлена формула дозволяє кількісно оцінити кожну можливу комбінацію одягу на основі об'єктивних критеріїв сумісності. Таким чином, система забезпечує персоналізовані рекомендації з урахуванням стилю, погодних умов і естетичної гармонії елементів гардеробу.

Висновки. Розроблений веб-застосунок дозволяє користувачам створити цифровий гардероб, автоматично класифікувати одяг та отримувати персоналізовані рекомендації на основі погодних умов і стилю. Використання великої мовної моделі для аналізу зображень забезпечує гнучкість та точність класифікації без необхідності складного навчання. Практичне значення роботи полягає у можливості використання розробленої системи як основи для консультацій щодо стилю одягу, персоналізованих рекомендацій та аналізу стилістичних уподобань користувачів.

Список літератури

1. Lu, C., Zhu, H., & Koniusz, P. (2024). DeepFashion2: Dataset Update with Enhanced Annotations and Expanded Categories. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, 491–498. https://service.tib.eu/ldmservice/dataset/deepfashion2

2. Shafiq, M., & Gu, Z. (2022). Deep Residual Learning for Image Recognition: A Survey. Applied Sciences, 12(18), 8972.

3. OpenAI. (2023). GPT-4 Technical Report. https://openai.com/research/gpt-4

4. Liu, Y., & Zhang, T. (2021). Style Classification and Outfit Recommendation Based on Fashion Dataset. Journal of Artificial Intelligence Research, 72(1), 35–49. https://doi.org/10.1007/s10462-021-09817-4

Гілета А.Р.
студентка 4 курсу ІКНІ, Національний університет "Львівська політехніка"
Науковий керівник к.т.н., асистент Антонів В.Я., кафедра АСУ
Національний університет "Львівська політехніка"

ІНФОРМАЦІЙНА СИСТЕМА АВТОМАТИЗОВАНОГО ТЕГУВАННЯ ТА КАТЕГОРИЗАЦІЇ ДОКУМЕНТІВ

Вступ. З кожним роком обсяг текстової інформації у цифровому вигляді стрімко зростає – у мережі щоденно публікуються тисячі (якщо не десятки тисяч) різноманітних новин, статей, публікацій та звітів. Подібний потік даних зумовлює необхідність в інструментах, які можуть автоматизувати обробку та аналіз текстів. Якщо колись потреба у передових технологіях поставала здебільшого у сферах, де постійно потрібні нові винаходи – науковій та військовій, – в той час як у повсякденному житті інформація поширювалась здебільшого у друкованому вигляді (газети, журнали, інші аналогові носії), то тепер процес проходження цифровізації проходять все дедалі більше повсякденні сфери, там, де ще донедавна домінували «друковані» документи і їхня ручна людська обробка. Відповідно до цього запиту, стає актуальним завдання створення рішень, які не просто зчитують текст, але й структурують його та проводять аналіз – тобто дозволяють систематизувати отриману інформацію. При цьому частина текстової інформації надходить у вигляді зображень або сканів.

З цього випливає ще одна проблема: окрім звичайної обробки текстів (plain text), існує потреба в опрацюванні зображень, які містять текст. Технологія розпізнавання тексту на зображеннях базується на OCR (optical character recognition) і є важливим засобом, який використовується у системах для архівування, пошуку та зберігання даних [1]. Але часто буває так, що сам по собі процес OCR не є достатнім для повноцінної роботи з текстом: необхідно застосовувати методи класифікації для організації інформації та витягувати ключові терміни для кращої індексації і пошуку. Тому особливо важливим кроком на шляху до діджиталізації є не лише «витягування» (extract) тексту із зображень, але й комбінування кількох методів обробки даних для досягнення кращої точності та ефективності.

В умовах наявності величезної кількості текстів, які потребують класифікації та позначення релевантними тегами, вирішення цієї задачі стає все більш необхідним та затребуваним. А оскільки інформаційні технології також не стоять на місці і активно розвиваються, логічним рішенням проблеми буде використання машинного навчання (machine learning) для навчання моделей категоризації тексту.

Для вирішення вищезазначених завдань актуальним є створення інформаційної системи, використовуючи передові інформаційні технології. Інформаційна система автоматизованого тегування та категоризації документів може значно допомогти опрацьовувати такі дані, як документи та зображення, які містять текст англійською мовою. Ця система дозволить завантажувати текстові документи та зображення (скріншоти), розпізнавати або витягувати з них текст, автоматично присвоювати їм категорію та теги, використовуючи ML-моделі, та виводити результати у зручному для людини вигляді. Це зекономить час на опрацювання документів і підвищить ефективність їх обробки, адже дозволить автоматизувати доволі рутинний процес і залучати людські ресурси лише на етапі завантаження документів та перевірки результатів роботи системи.

Постановка задачі. Метою роботи є створення та впровадження алгоритмів, методів, засобів і ML-моделей для інформаційної системи автоматизованого тегування та категоризації документів англійською мовою. Система повинна забезпечувати обробку і аналіз текстів, проводити класифікацію документів за змістом на основі попередньо натренованих моделей машинного навчання, автоматично визначати релевантні ключові теги тексту та представляти результати роботи системи у наочному та зручному для користувача вигляді – у формі графіка «хмари слів» (“Word Cloud”).

З метою збору даних для дотреновування моделей машинного навчання задля запобігання їхньої деградації, користувач зможе також надати свій дозвіл на збір анонімізованої

завантаженої інформації за дотриманням усіх вимог та правил GDPR [2]. Об'єктом дослідження є процеси розпізнавання символів на зображеннях; обробки, аналізу та класифікації документів, які містять текст англійською мовою. Предметом дослідження є моделі машинного навчання та алгоритми для категоризації й тегування англомовних текстів, засоби їхньої інтеграції у інформаційну веб-систему для забезпечення комфортної взаємодії з кінцевим користувачем.

Важливою перевагою системи серед аналогів у галузі інформаційних послуг, стане саме надання зручного порталу для опрацювання документів, а не надання API, як це зазвичай роблять конкуренти та аналогічні системи.

Отже, область застосування цієї інформаційної системи охоплює широкий спектр сценаріїв використання: від особистого використання дослідниками, студентами чи журналістами до інтеграції системи в корпоративні або інформаційні платформи для прискорення обробки текстових даних.

Основний матеріал. Реалізація інформаційної системи автоматизованого тегування та категоризації документів вимагає створення програмних рішень для збору, опрацюванням зображень, обробки та аналізу документів з подальшою їхньою категоризацією, тегуванням та виведенням результатів. Кожен з етапів автоматизації має свої особливості, що вплинуло на вибір алгоритмів, бібліотек і програмну реалізацію компонентів системи. Нижче наведено ключові виклики та реалізовані рішення, що формують основу запропонованої системи.

Перший виклик – необхідно забезпечити зручний веб-інтерфейс для інтеграції всіх функцій інформаційної системи (завантаження файлів, обробка, класифікація, візуалізація) та об'єднання логіки застосунку в одному середовищі. Його рішенням є реалізація односторінкового веб-застосунку зі зручним UI. Система містить компонент інтерфейсу користувача, який створений за допомогою бібліотеки Streamlit (генерує фронтенд засобами HTML/JS, але пишеться на Python). Для створення інтерактивних елементів UI та дизайну окремих елементів інтерфейсу використано HTML-розмітку та CSS-стилі.

Під час розробки веб-застосунку інформаційної системи виникла потреба забезпечити злагоджену роботу численних компонентів системи – від розпізнавання тексту до класифікації, візуалізації результатів і взаємодії з базою даних, – у межах єдиного середовища з простою архітектурою та мінімальними витратами на підтримку. Її рішенням стала реалізація серверної частини з монолітною архітектурою. Такий підхід передбачає інтеграцію всіх розроблених модулів в єдине програмне середовище, що працює як односторінковий динамічний веб-застосунок. Це дозволяє досягти високої узгодженості між модулями, зменшити складність процесів розробки і технічної реалізації. Для цього було реалізовано описану серверну частину на мові Python, яка керує всіма створеними моделями та взаємодіє із базою даних, яка реалізована на основі MS SQL Server.

Ще однією задачею, яка частково була описана вище, є представлення текстової інформації у вигляді зображень, які не підлягають автоматичній обробці без попереднього розпізнавання. Для повноцінного аналізу таких джерел необхідно точно витягнути текстовий вміст. Рішенням цієї проблеми стала реалізація модуль розпізнавання тексту з графічних файлів за допомогою технології OCR (optical character recognition). Після завантаження зображення від користувача виконується попередня обробка (за допомогою бібліотеки OpenCV – перетворення в чорно-білий формат, підвищення різкості, бінаризація, морфологічні операції очищення фону), після чого відбувається розпізнавання тексту зображення з використанням інструменртів бібліотеки pytesseract. Додатково у компоненті використовують бібліотеки NumPy – для роботи з масивами зображень у вигляді матриць пікселів, та PIL (Pillow) – для роботи із зображеннями у форматі Image. Створений компонент підтримує розпізнавання англійської мови та налаштування для збереження пробілів між словами, що особливо важливо для точності подальшої обробки.

Для реалізації ефективної категоризації текстів виникла необхідність у використанні точних та швидкодіючих моделей машинного навчання, які могли б забезпечити надійну класифікацію даних [3]. Рішенням цієї проблеми стало попереднє тренування дев'ятьох різних моделей машинного навчання з використанням бібліотек scikit-learn, catboost, xgboost та lightgbm, результати яких порівнювалися за відповідними метриками точності, повноти та F1-міри на тестових вибірках. На основі аналізу отриманих результатів було відібрано три моделі з

найкращими показниками. Після чого ці моделі було збережено у файловій системі для подальшого використання. Шляхи до збережених моделей заносяться до бази даних. Це дозволяє зручно обирати одну з них під час роботи з інформаційною системою. У результаті було реалізовано компонент класифікації текстів на серверній частині. Саме тут завантажуються попередньо треновані моделі та виконується класифікації тексту користувача, після чого отримується відповідна категорія тексту, яка далі передається у модуль графічного інтерфейсу користувача.

Автоматичне виділення змістовних ключових слів із тексту також є складною задачею, яка постала при реалізації інформаційної системи, оскільки потребує розуміння лексичного контексту, граматичної структури й семантичної релевантності слів без втручання користувача. Рішенням цієї задачі є створений модуль автоматичного тегування інформаційної системи. Він поєднує класичні методи обробки природної мови із сучасними підходами та інструментами. На початковому етапі текст очищується і обробляється з використанням бібліотеки spaCy, яка виконує токенізацію, лематизацію та визначає частини мови. Наступним етапом є виділення з-поміж лематизованих слів кандидатів на ключові слова, для яких далі обчислюються семантичні векторні представлення (ембеддінги) за допомогою бібліотеки torch та моделі BertModel з бібліотеки transformers [4]. Тоді на основі косинусної подібності (інструмент cosine_similarity із бібліотеки scikit-learn) обираються найрелевантніші слова, що і формують підсумковий набір тегів.

Після категоризації та тегування тексту користувачу необхідно надати зручне візуальне представлення змісту документа. Текстова інформація може бути громіздкою для сприйняття, тому ефективною формою подачі є графічна інтерпретація ключових слів. Для цього у системі реалізовано окремий модуль візуалізації, що генерує хмару слів, яка є наочним представленням найчастіше вживаних слів у тексті документа. Реалізація відбувається на основі інструментів з бібліотек wordcloud (генерація хмари слів) та matplotlib (графічне виведення хмари слів).

Висновки. Розроблена інформаційна система завдяки використанню попередньо натренованим моделям машинного навчання та інструментів обробки природної мови (OCR, BERT, spaCy) дозволяє автоматизовано, якісно та швидко проводити категоризацію й тегування завантажених текстових документів, що значно скорочує час проведення класифікації та структурування текстової інформації.

Система є затребуваною як серед індивідуальних користувачів, які потребують швидкого розпізнання тексту на зображеннях, чи класифікації та аналізу текстової інформації; так і для організацій, що обробляють великі обсяги документів і прагнуть автоматизувати процеси своєї аналітики, таких як: медіакомпаній, новинних агентств, бібліотек, архівів, освітніх закладів та контент-маркетингових агентств. Широкий спектр застосування системи підтверджує її практичну цінність.

Список літератури

1. Greif, G., Griesshaber, N., & Greif, R. (2025). *Multimodal LLMs for OCR, OCR Post-Correction, and Named Entity Recognition in Historical Documents*. arXiv Cornell University. https://doi.org/10.48550/arXiv.2504.00414
2. Aberkane, A., vanden Broucke, S., & Poels, G. (2023). *Factors related to GDPR compliance promises in privacy policies: A machine learning and NLP approach*. International Journal of Information Security and Privacy (IJISPM), 13(2). https://doi.org/10.12821/ijispm130202
3. Allam, H., Makubvure, L., Gyamfi, B., Nyarko Graham, K., & Akinwolere, K. (2025). *Text Classification: How Machine Learning Is Revolutionizing Text Categorization*. Information, 16(2), 130. https://doi.org/10.3390/info16020130
4. Kataria, V. (2025). *A hybrid architecture for multi-category text classification using BERT and graph embeddings*. International Journal of Scientific Research in Engineering and Management (IJSREM), 9(5), 1–9. https://doi.org/10.55041/IJSREM46999

Завалій Т.І.[(1)], Шаховська Н.Б.[(2)]
(1) аспірант 3 курсу ІКНІ, (2) д.т.н., професор, ректор
Національний університет "Львівська політехніка"
Науковий керівник д.т.н., професор Шаховська Н.Б., каф. СШІ,
Національний університет "Львівська політехніка"

ОПЕРАТОРИ ОПРАЦЮВАННЯ ПОТОКОВИХ ДАНИХ В ЗАДАЧІ МОДЕЛЮВАННЯ РУХУ ПІШОХОДА

Вступ. Алгоритми, придатні для опрацювання потокових даних, застосовують в таких галузях як спортивна аналітика (відстеження рухів спортсменів), навігаційні і транспортні системи (логістика, безпілотні транспортні засоби), медична аналітика (відстеження стану пацієнта), моніторинг хмарної інфраструктури, торгівля на фондовому ринку тощо. А з появою дешевих MEMS сенсорів, набув нового прикладного значення аналіз потоків даних з мобільних пристроїв з багатьма сенсорами: прискорення та орієнтації (IMU), геолокації (GNSS), магнітометром, барометром. Поєднання даних з кількох джерел в режимі реального часу дозволяє уточнювати положення об'єкту в просторі і мінімізувати похибку окремого сенсора [1].

Постановка задачі. Об'єктом даного дослідження є процеси опрацювання потоків даних з метриками і координатами рухомого об'єкту. Предмет дослідження – модель руху тіла на основі потокових даних сенсорів смартфону. Мета дослідження полягає в розробленні операторів опрацювання потоків прискорення і орієнтації в задачі моделювання руху, та їх реалізація в середовищі паралельних потокових обчислень (Apache Flink). Модель руху пішохода опишемо стандартною залежністю швидкості $v(t)$ від прискорення $a(t)$:

$$v(t) = v(t-1) + \int_{t-1}^{t} a(t)dt \qquad (1)$$

Проблему дрейфу при подвійному інтегруванні прискорення для отримання пройденої відстані розв'язано шляхом побудови LSTM моделі для прогнозування швидкості руху з похибкою *MAE=0.087 м/с* і R^2*=0.83* на тестових даних [2].

Основний матеріал. Операції над потоками поділяються на два основні типи: без збереження стану (stateless) і зі збереженням стану (statefull). До першого типу належать базові оператори, що застосовуються до кожного елементу потоку x_t [3]: *Map* – перетворення кожного елементу потоку, $f: x_t \mapsto y_t$; *FlatMap* – перетворення елементу потоку у кілька результуючих, $f: x_t \mapsto y_t^1, \dots y_t^k$; *Filter* – вилучення елементів потоку за предикатом, $f: x_t \mapsto x_t \; if \; P(x_t)$; *Union* – об'єднання двох потоків, $U = S \cup T$; *KeyBy* – розщеплення потоку даних на підпотоки за значенням ключа, $f: x_t \mapsto (k, x_t)$; *Window* – формування інтервалів часу W_i, за якими вибираються елементи з потоку, $W_i = [t_i, t_i + \Delta t]$.

До операторів зі збереженням стану, наприклад, належать [3]: *Join* – злиття двох потоків за значенням ключа або часовою міткою, $f: (x_t, z_t) \mapsto h(x_t, z_t)$; *Apply* – застосування функції перетворення до всіх елементів у вікні, $f: x_t^W \mapsto y_t^W$; *Reduce* – послідовне застосування агрегатної функції у вікні, $y^W = \sum_{t \in W} x_t$.

На основі базових операторів, введемо додаткові оператори, специфічні для нашої задачі:

- *LowPass* – низькочастотне фільтрування потоку (сигналу) з заданою граничною частотою ω_c у вікні W_t, $\widetilde{W}_t = \mathcal{L}_{w_c}^n(W_t) = filtfilt(a, b, W_t)$.
- *PolyFilt* – поліноміальний фільтр Savitsky-Golay для вікна довжиною $2k$+1 і степенем полінома n, $\tilde{x}_t = S_{k,n}(W_t) = \sum_i (-k^k c_i x_{t+i})$.
- *ToENU* – злиття потоків прискорення та орієнтації із застосуванням матриці обертання R_t для перетворення системи координат з локальної в глобальну, $\vec{a}_t^{local} \mapsto R_t \cdot \vec{a}_t^{local} = \vec{a}_t^{ENU}$.
- *Enrich* – обчислення додаткових ознак для LSTM моделі, $\phi: x_t \mapsto [x_t, f_1(x_t), \dots, f_n(x_t)]$.
- *Predict* – прогнозування швидкості руху на основі даних сенсора прискорення і додаткових ознак (вектор x_t) у вікні довжиною w, $x_t \mapsto \hat{y}_t = \mathcal{F}_\theta[x_{t-w+1}, x_{t-w+2}, \dots, x_t]$.

Для реалізації описаних операторів було використано Python, Kafka, Apache PyFlink, Opensearch Dashboards, а також такі бібліотеки як Numpy, SciPy, Pytorch, Quix Streams.

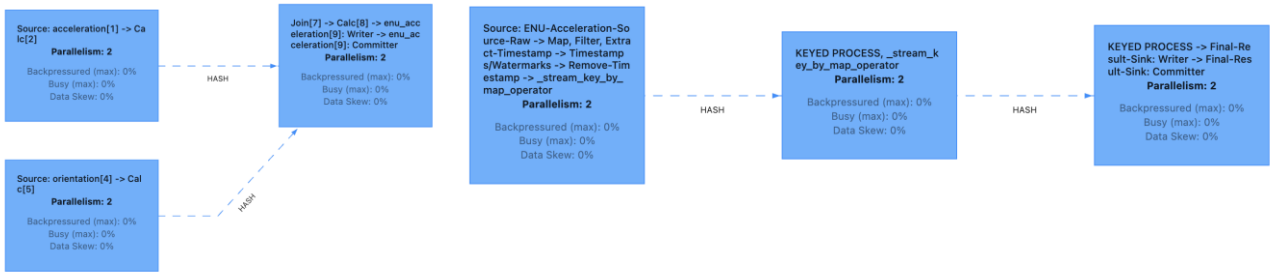


Рисунок 1 – Оператори злиття *ToENU* (зліва) та прогнозування *Predict* (справа) в Apache Flink

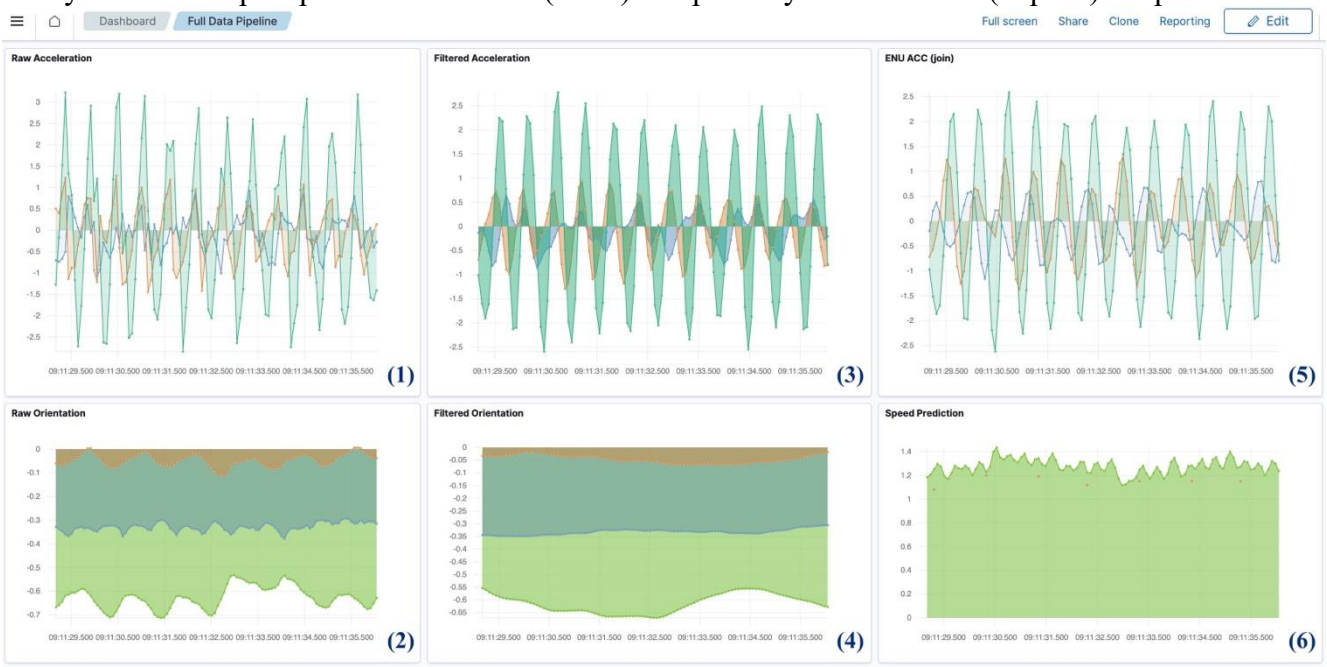


Рисунок 2 – Результати опрацювання потоків подані в Opensearch Dashboards

Цикл опрацювання потоків зображено на рис. 2. Зверху вниз і зліва направо: 1) сирі дані прискорення по осям *x*, *y*, *z* з частотою 53Hz; 2) сирі дані орієнтації (pitch, roll, yaw) з частотою 53Hz; 3) фільтрування прискорення оператором *LowPass*; 4) фільтрування орієнтації оператором *PolyFilt*; 5) злиття прискорення та орієнтації оператором *ToENU*; 6) прогнозування швидкості оператором *Predict* у ковзаючому вікні в 2 секунди.

Висновки. Подано розв'язок задачі прогнозування швидкості руху на основі потокових даних сенсорів смартфону. Перевагою реалізації введених операторів в середовищі Apache PyFlink є швидкодія, масштабованість, та доступ до всіх python-бібліотек для машинного навчання та аналізу даних. Недоліком є висока складність інтеграції JVM-середовища Apache Flink з виконуваним середовищем Python.

Список літератури

1. Asaad S., Maghdid H., *A Comprehensive Review of Indoor/Outdoor Localization Solutions in IoT era*, TechRxiv, – 2021. – Режим доступу: https://www.techrxiv.org/doi/full/10.36227/techrxiv.15138609 (дата звернення: 1.05.2025).

2. Завалій Т.І., "Моделювання руху пішохода на основі даних сенсорів смартфону", Вісник Хмельницького національного університету, серія: технічні науки, №6, Т.1, – 2024, – С. 273-278.

3. Apache Flink. Operators. – Електронний ресурс. – Режим доступу: https://nightlies.apache.org/flink/flink-docs-release-1.20/docs/dev/datastream/operators/overview (дата звернення: 1.05.2025).

Керницький О.Б.(1), Керницький А.Б.(2)
1) Аспірант ІІІ року АСУ ІКНІ Національний університет "Львівська політехніка"
Науковий керівник д.т.н., проф. Теслюк В.М. кафедра АСУ, Національний університет "Львівська політехніка"
2)Доктор філософії, доцент кафедри САП, Національний університет "Львівська політехніка"

РОЗРОБЛЕННЯ СПЕЦИФІКАЦІЙ ТА ВИМОГ В ПРОЦЕСІ РЕІНЖИНІРИНГУ ІТ-ПРОЕКТІВ З ВИКОРИСТАННЯМ ШІ

Вступ. Зростаюче поширення застарілих ІТ-систем та потреба в їх реінжинірингу та побудові нових систем для задоволення сучасних бізнес-вимог і технологічних досягнень є актуальною проблемою для багатьох організацій. Архаїчні системи нерідко характеризуються високими витратами на обслуговування, обмеженою масштабованістю та вразливостями безпеки. Необхідність модернізації цих систем стає дедалі. Дослідження показують, що значна частина ІТ-бюджетів витрачається на підтримку застарілих систем, що обмежує можливості для інновацій. Крім того, процеси реінжинірингу стикаються з викликами, пов'язаними з оцінкою зусиль та ресурсів. Залежність від застарілих систем, незважаючи на їхні недоліки, часто зумовлена значними вже здійсненими інвестиціями та відчутними ризиками, пов'язаними з міграцією. Ця інерція підкреслює критичну потребу в стратегіях модернізації.

Потенціал штучного інтелекту (ШІ) для революціонізування різних аспектів розробки програмного забезпечення, включаючи інженерію вимог та модернізацію систем, є значним. Огляд існуючих досліджень підкреслює зростаючий інтерес до застосування ШІ для автоматизації та покращення процесу вилучення вимог [1]. ШІ стає трансформаційною силою в розробці програмного забезпечення. Його здатність автоматизувати завдання, аналізувати великі набори даних та надавати інтелектуальні висновки робить його перспективним інструментом для вирішення складнощів реінжинірингу ІТ.

Зрозумілі та вичерпні вимоги є основою успішної реалізації проекту. Неоднозначні або неповні вимоги можуть призвести до значних переробок, затримок та збільшення витрат. Якість початкових вимог безпосередньо впливає на успіх усього проекту реінжинірингу.

При роботі з розробленням специфікацій та вимог в процесі реінжинірингу ІТ-проектів застосовується класичні підходи для отримання вхідної інформації. Мною уже було розроблено метод синтезу специфікацій при реінжинірингу ІТ-проектів [2]. Він описує покроково необхідні дії, такі як, збір даних про існуючу систему (збір вимог, аналіз документів, реверсивна розробка, опитування спеціалістів), аналіз отриманих результатів та синтез попереднього варіанту вимог, розроблення та тестування моделі чорного ящика для існуючої системи, аналіз результатів тестування, та за необхідності корегування вимог [3].

Реінжиніринг застарілого програмного забезпечення часто передбачає боротьбу зі складними, недокументованими архітектурами та застарілими базами, особливо коли первісні розробники вже недоступні. Ця відсутність ясності становить значну перешкоду для точного відображення поточної функціональності та передбачуваної поведінки системи. Ця проблема ускладнюється тим, що застарілі системи часто мають низьку якість документації або взагалі її відсутність, що значно збільшує зусилля на їх аналіз. Накопичений за роки модифікацій та швидких виправлень "технічний борг" у застарілих системах ще більше ускладнює розуміння та вилучення первісних вимог.

Постановка задачі. Об’єктом дослідження є сучасні ІТ-проєкти, що реалізуються в умовах швидкоплинних технологічних змін. Предметом дослідження виступає процес реінжинірингу ІТ-проєктів із застосуванням методів і засобів штучного інтелекту (ШІ). Метою є обґрунтування доцільності інтеграції ШІ в процеси реінжинірингу та розробка підходу до його ефективного використання в управлінні життєвим циклом ІТ-проєктів. Актуальність дослідження зумовлена необхідністю адаптації підходів до управління ІТ-проєктами відповідно до нових можливостей, які надає ШІ.

Основний матеріал. Відповідно до сформульованої мети дослідження, далі розглянуто особливості використання засобів ШІ в реінжинірингу сучасних ІТ-проєктів. ШІ може

автоматизувати збір вимог з різних джерел, включаючи існуючу документацію, код та взаємодії зі стейкхолдерами. Інструменти на базі ШІ можуть автоматично вилучати дані з різноманітних джерел, таких як електронні листи, PDF-файли, репозиторії коду та існуючі специфікації, значно зменшуючи ручну роботу, пов'язану з початковим збором вимог. Методи NLP дозволяють ШІ обробляти текстову інформацію з цих джерел для виявлення потенційних вимог. ШІ може обробляти ширший спектр форматів даних та більші обсяги даних, ніж ручні методи, потенційно виявляючи вимоги, які могли б бути пропущені.

Використання методів обробки природної мови (NLP) та машинного навчання (ML) для аналізу та розуміння вимог, виражених природною мовою, є ключовим. Особливо методи NLP показали свою ефективність у виявленні вимог з неструктурованих текстів, що є типовим для документації застарілих систем [4]. Алгоритми NLP можуть аналізувати вимоги, виражені природною мовою, для виявлення ключових сутностей, взаємозв'язків та намірів користувачів. Методи ML можуть використовуватися для класифікації вимог за різними категоріями (наприклад, функціональні, нефункціональні) та для виявлення потенційних проблем, таких як неоднозначність або несумісність. Застосування NLP та ML забезпечує більш структурований та автоматизований аналіз вимог, що призводить до покращення чіткості та зменшення суб'єктивності порівняно з чисто ручним аналізом.

Здатність ШІ виявляти неявні вимоги, невідповідності та неоднозначності в існуючих системах та потребах стейкхолдерів є цінною. Аналізуючи відгуки користувачів, системні журнали та історичні дані, ШІ може виводити неявні вимоги, які стейкхолдери можуть явно не формулювати. ШІ також може перехресно зіставляти вимоги з різних джерел для виявлення невідповідностей та позначати неоднозначну мову, яка може призвести до неправильного розуміння. Виявлення неявних вимог може призвести до більш повного розуміння потреб користувачів, а виявлення невідповідностей та неоднозначностей на ранньому етапі може запобігти дорогим переробкам на пізніших етапах проекту реінжинірингу.

ШІ також може бути партнером для спільного мозкового штурму, стимулюючи креативні ідеї та виявляючи нові можливості. Інструменти ШІ можуть виступати як партнери для мозкового штурму, пропонуючи потенційні вимоги, функції та функціональні можливості на основі аналізу існуючих систем, ринкових тенденцій та очікувань користувачів. Це може допомогти у виявленні інноваційних рішень та розкритті раніше неврахованих можливостей для реінжинірингової системи. Здатність ШІ обробляти та синтезувати інформацію з різних джерел може забезпечити ширшу перспективу та потенційно призвести до більш інноваційних та конкурентоспроможних реінжинірингових ІТ-проектів.

Подивимося, чи існують готові інструменти та розроблені методи на базі ШІ для генерації специфікацій. Використання ШІ для генерації початкових чернеток специфікацій, включаючи функціональні специфікації, історії користувачів та технічні вимоги. ШІ, особливо генеративні моделі ШІ, може створювати початкові чернетки специфікацій, історій користувачів та навіть технічних вимог на основі виявлених вимог та розуміння існуючої системи. Ці згенеровані ШІ чернетки можуть слугувати відправною точкою для подальшого вдосконалення експертами. Автоматизація генерації початкових чернеток специфікацій може значно скоротити час та зусилля, необхідні на ранніх етапах реінжинірингу, дозволяючи командам зосередитися на валідації та складніших аспектах. Аналіз ринку готових рішень показує, що існує змагання в різних підходах та методах використання ШІ для написання специфікацій.

Роль ШІ у забезпеченні чіткості та узгодженості різних частин специфікацій є важливою. Інструменти на базі ШІ можуть аналізувати згенеровані специфікації для забезпечення узгодженості термінології, форматування та рівня деталізації в різних розділах. Вони також можуть виявляти потенційні протиріччя або перекриття між різними вимогами, допомагаючи створити більш узгоджений та уніфікований документ специфікації. Покращена чіткість та узгодженість специфікацій зменшують ймовірність неправильного розуміння з боку команди розробників та сприяють створенню більш надійної та зручної в обслуговуванні реінжинірингової системи.

Допомога ШІ у вдосконаленні та валідації специфікацій шляхом аналізу їх на наявність невідповідностей та неоднозначностей є значною. ШІ може безперервно аналізувати та вдосконалювати специфікації, виявляючи залишки невідповідностей, неоднозначностей або області, де потрібна додаткова деталізація. Цей ітеративний процес, підтримуваний зворотним зв'язком від ШІ, допомагає покращити загальну якість та повноту специфікацій. Можливості безперервного аналізу та вдосконалення ШІ можуть призвести до більш точних та вичерпних специфікацій, мінімізуючи потребу в переробці під час фази впровадження.

ШІ може аналізувати згенеровані специфікації для автоматичного створення тестових випадків, тестових сценаріїв та іншої супутньої документації, ще більше спрощуючи процес реінжинірингу. Ця автоматизація може значно скоротити час та зусилля, необхідні для тестування та документування. Автоматизація генерації тестових випадків забезпечує краще покриття тестами та може призвести до вищої якості реінжинірингової системи шляхом виявлення потенційних дефектів на ранніх етапах циклу розробки.

Отже трансформація застарілих вимог за допомогою ШІ є можливою та виглядає дуже перспективною. ШІ може аналізувати застарілі кодові бази та вилучати вбудовану бізнес-логіку та вимоги. Цей процес вилучення часто вимагає методів, що дозволяють 'відобразити' архітектуру системи, щоб зрозуміти її структуру та поведінку, що є важливим для точного відновлення вимог. Інструменти аналізу коду на базі ШІ можуть сканувати застарілий код для виявлення закономірностей, залежностей та бізнес-правил, які можуть бути явно не задокументовані. Це допомагає зрозуміти функціональність існуючої системи та виявити приховані вимоги. Здатність ШІ розшифровувати складний застарілий код може подолати проблему відсутності або застарілої документації, забезпечуючи більш точне розуміння внутрішньої роботи системи.

Використання ШІ для рефакторингу та перекладу коду зі застарілих мов на сучасні є важливим. ШІ, особливо генеративний ШІ, може допомогти у рефакторингу застарілого коду для покращення його читабельності, зручності обслуговування та продуктивності. ШІ також може перекладати код зі старих мов, таких як COBOL або Fortran, на сучасні мови, такі як Java або Python, полегшуючи міграцію на новіші платформи. Автоматизація рефакторингу та перекладу коду може значно скоротити час та витрати, пов'язані з модернізацією застарілих систем, та зменшити ризики, пов'язані з використанням застарілих технологій та навичок.

ШІ відіграє важливу роль у виявленні та пропонуванні покращень продуктивності та зручності обслуговування застарілих систем. Алгоритми ШІ можуть аналізувати застарілий код для виявлення вузьких місць продуктивності, вразливостей безпеки та областей високої складності, пропонуючи оптимізації та покращення для підвищення загальної якості системи. Проактивно виявляючи та пропонуючи покращення, ШІ може допомогти у створенні більш надійної та ефективної реінжинірингової системи, яка усуває недоліки застарілої системи.

Генерація документації для застарілих систем за допомогою ШІ є цінною можливістю. ШІ може автоматично генерувати документацію для застарілих систем, аналізуючи код, визначаючи ключові функціональні можливості та створюючи зрозумілі описи. Це вирішує поширену проблему відсутності або застарілої документації, полегшуючи розуміння та обслуговування системи. Згенерована ШІ документація може значно скоротити зусилля, необхідні для передачі знань та адаптації нових членів команди, які працюють над реінжинірингговою системою.

Переваги використання ШІ є значними, проте існують певні виклики використання ШІ в інженерії вимог для реінжинірингу ІТ-проектів. Переваги включають підвищення ефективності та автоматизацію, покращення точності та узгодженості, покращення співпраці, покращення прийняття рішень, зниження витрат та скорочення часу виходу на ринок, посилення безпеки та відповідності нормативним вимогам.

Виклики включають високі початкові витрати та вимоги до навчання, залежність від якості даних, складність інтеграції із застарілими системами, етичні проблеми та потенційні упередження, ризик скорочення робочих місць, необхідність людського контролю.

Переваги ШІ в цьому контексті є значними, пропонуючи істотні покращення в ефективності, якості та вартості. Однак виклики, особливо пов'язані з якістю даних, складністю

інтеграції, потребують ретельного управління для забезпечення успішного та відповідального впровадження ШІ в реінжиніринг ІТ.

ШІ може автоматизувати створення розділів у традиційних документах, таких як SRS та BRD (Software Requirements Specification - SRS, Business Requirements Document - BRD). ШІ може допомогти у створенні та вдосконаленні історій користувачів.

Штучний інтелект може аналізувати існуючу застарілу документацію та код для вилучення ключових функціональних можливостей та бізнес-правил, які потім можуть бути використані для створення історій користувачів або інших сучасних форматів вимог, придатних для гнучких методологій розробки.

ШІ також може виступати як міст між традиційними, часто більш формальними, підходами, що передбачають велику кількість документації, та сучасними, більш ітеративними та орієнтованими на користувача гнучкими методологіями. Це може полегшити модернізацію застарілих систем шляхом перетворення їхніх вимог у формати, які краще підходять для сучасних практик розробки.

Висновки. ШІ надає значну можливість революціонізувати створення специфікацій та вимог у проектах реінжинірингу ІТ, пропонуючи істотні переваги щодо ефективності, точності та зниження витрат.

Інтеграція ШІ з традиційними та сучасними методологіями інженерії вимог може призвести до більш ефективних та спрощених процесів реінжинірингу, що в кінцевому підсумку сприяє успішній модернізації критичної ІТ-інфраструктури.

Потрібні подальші дослідження для вирішення виявлених проблем та для розкриття повного потенціалу ШІ в цій галузі, особливо у розробці надійних фреймворків оцінки та етичних принципів для інженерії вимог за допомогою ШІ.

Список літератури

1. Najafi Mohsenabad, H., & Tut, M. A. (2024). Optimizing Cybersecurity Attack Detection in Computer Networks: A Comparative Analysis of Bio-Inspired Optimization Algorithms Using the CSE-CIC-IDS 2018 Dataset. Applied Sciences, 14(3), 1044. https://doi.org/10.3390/app14031044

2. Kernytskyy, O., Teslyuk, V. (2023). The Synthesis Method For Specifications And Requirements In The Process Of IT Project Reengineering. Ukrainian Journal of Information Technology, 5(2), 01–08. https://doi.org/10.23939/ujit2023.02.001

3. Kernytskyy, O.B., Kernytskyy, A., & Teslyuk, V. (2023). The Synthesis Method for Specifications and Requirements in the Process of IT Project Reengineering. 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT), 1-4. https://doi.org/10.1109/CSIT61576.2023.10324175

4. Guo, J.L.C., Steghöfer, JP., Vogelsang, A., Cleland-Huang, J. (2025). Natural Language Processing for Requirements Traceability. In: Ferrari, A., Ginde, G. (eds) Handbook on Natural Language Processing for Requirements Engineering. Springer, Cham. https://doi.org/10.1007/978-3-031-73143-3_4

Штогрінець Б. В.
аспірант, кафедра автоматизованих систем управління,
Національний університет "Львівська політехніка"
Науковий керівник д-р техн. наук, професор Цмоць І. Г., кафедра автоматизованих систем управління, Національний університет "Львівська політехніка".

ПРОБЛЕМНО-ОРІЄНТОВАНА КРИПТОГРАФІЧНА СИСТЕМА НЕЙРОМЕРЕЖЕВОГО ЗАХИСТУ ДАНИХ

Вступ. Розвиток сучасних інформаційних технологій, зокрема в галузі керування безпілотними системами та мобільними робототехнічними платформами, актуалізує питання захисту даних у реальному часі. Існуючі підходи до криптографічного захисту не забезпечують необхідної швидкодії, гнучкості та адаптивності, особливо в умовах обмежених ресурсів бортових систем.[1] У таких умовах доцільним є впровадження нейромережевих методів шифрування-дешифрування з таблично-алгоритмічною реалізацією, що здатні забезпечити високу криптостійкість і реальновчасну обробку інформації на рівні апаратних засобів.[2]

Постановка задачі. Об'єкт дослідження – розроблення проблемно орієнтованої системи криптографічного захисту даних. Предмет дослідження – методи та засоби побудови нейромережевого криптографічного захисту в режимі реального часу. Метою дослідження є розроблення ефективної системи, що базується на автоасоціативних нейромережах прямого поширення, реалізованих з використанням таблично-алгоритмічного обчислення скалярного добутку. Основними завданнями є:

- удосконалення методу обчислення скалярного добутку для роботи з ваговими коефіцієнтами у форматі з плавучою комою;
- розроблення базової структури пристрою; створення потокової архітектури шифрування/дешифрування; реалізація прототипу на FPGA.

Основний матеріал. Метод нейромережевого шифрування даних побудовано на використанні автоасоціативної нейромережі прямого поширення з попередньо обчисленими ваговими коефіцієнтами. Архітектура такої мережі визначається кількістю нейронних елементів, кількістю входів та їх розрядністю. Ключ до системи включає маску, матрицю вагових коефіцієнтів та конфігурацію мережі. Шифрування виконується через XOR-маскування, скалярне множення векторів з використанням вагових коефіцієнтів, реалізованих у таблично-алгоритмічному вигляді.[3]

Особливістю обчислення скалярного добутку є використання багатоопераднього табличного підходу, де попередньо формуються таблиці макрочасткових добутків. Їх зчитування виконується за адресами, які формуються на основі розрядних зрізів вхідного вектора. Для скорочення часу обчислення застосовується розпаралелення та зсувна схема підсумовування. Скалярний добуток виконується у два етапи: підготовка таблиць та зчитування й акумулювання добутків.

Розроблено базову структуру таблично-алгоритмічного пристрою обчислення скалярного добутку, яка складається з регістрів введення даних, блоку пам’яті, суматора, шинних формувачів та мікроконтролера. У пристрої реалізовано два режими: запис таблиць та обчислення. Запис таблиць забезпечується через окремі керуючі сигнали, а обчислення виконується послідовно з урахуванням циклів тактування.[4]

Структуру проблемно орієнтованої підсистеми нейромережевого шифрування даних наведено на рис. 1, де МК – мікроконтролер, ВМ – вузол маски, НМ – нейромережа, МД – макрочастковий добуток, Рг – регістр, См – суматор.

суматор.

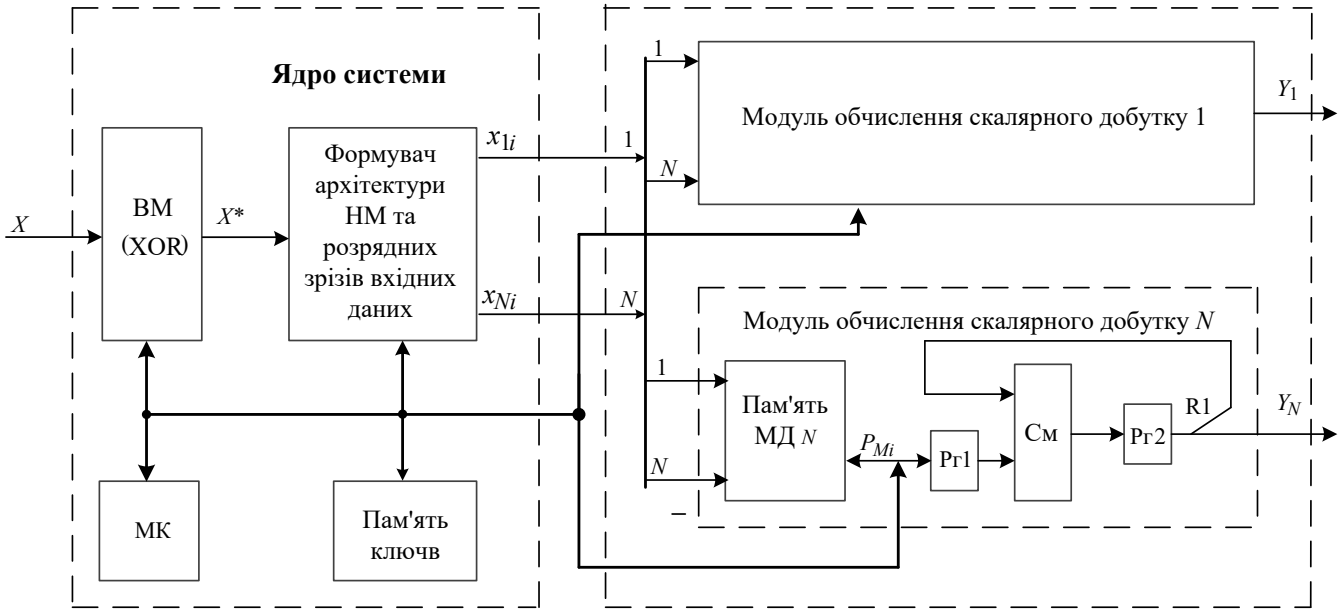


Рисунок 1 - Структура проблемно орієнтованої підсистеми нейромережевого шифрування даних

Структура проблемно орієнтованої підсистеми нейромережевого криптографічного дешифрування даних переважно відповідає структурі підсистеми нейромережевого шифрування даних, яку наведено на рис.1.

Реалізація системи на базі FPGA EP3C16F484C6 засвідчила високу продуктивність (160 нс на вектор) при невеликій кількості задіяних логічних елементів (3053) і регістрів (745), що свідчить про доцільність впровадження такого підходу в бортові системи. Табличо-алгоритмічний підхід у поєднанні з автоасоціативними нейромережами забезпечує високу швидкодію, криптостійкість, адаптивність до типу вхідних даних, гнучкість масштабування та ускладнення криптоаналітичних атак. [5]

Принципову схему спеціалізованих апаратних засобів підсистеми нейромережевого шифрування даних наведено на рис. 2. Входи блоку XOR_Mask1_4_2: *X*[7..0] – вхідні дані; *Clk* – вхід синхронізації завантаження вхідних даних; *X_Mask*[7..0] – 8-бітна маска. На виході цього блоку формується *N* векторів розрядністю *m*. Синхронізацію реалізовано за переднім фронтом імпульсів *Clk*.

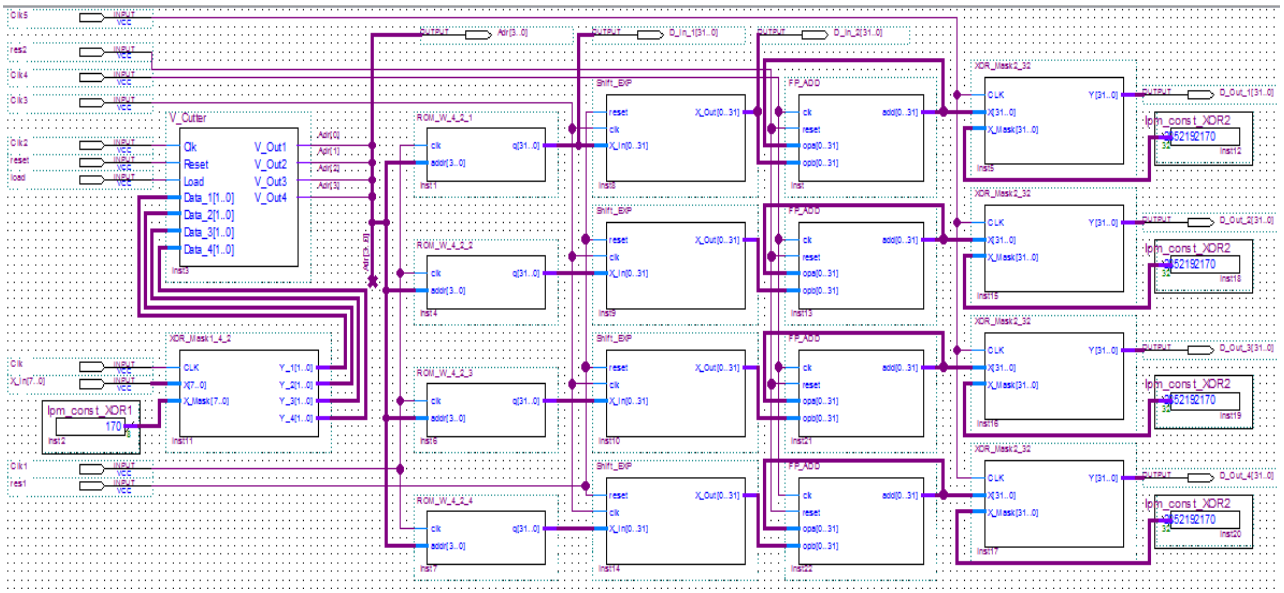


Рисунок 2 - Принципова схема спеціалізованих апаратних засобів підсистеми нейромережевого шифрування даних

Отримані результати підтверджують ефективність запропонованого методу нейромережевого шифрування даних, реалізованого на базі FPGA. Використання лише 3053 логічних елементів (менше 20 % від доступних ресурсів кристала) та 745 регістрів свідчить про раціональне використання апаратної бази, що є критично важливим для бортових систем з обмеженими ресурсами. Швидкодія на рівні 160 наносекунд для оброблення одного вектора вхідних даних демонструє доцільність застосування таблично-алгоритмічного підходу до реалізації нейроподібних елементів, що забезпечує виконання криптографічних операцій у режимі реального часу.

Висновки. Розроблено проблемно-орієнтовану систему нейромережевого криптографічного захисту даних у реальному часі, що поєднує апаратно-програмні засоби, автоасоціативні нейромережі прямого поширення, адаптивну структуру та вдосконалений таблично-алгоритмічний метод обчислення скалярного добутку. Вона забезпечує високу швидкодію, ефективність і криптографічну стійкість у межах обмежених ресурсів. Запропонований підхід може бути основою для подальших досліджень і впроваджень у сфері безпечної обробки даних у вбудованих системах.

Список літератури

1. Awad, A. I., Furnell, S., Paprzycki, M., & Sharma, S. K. (Eds.). (2021). Security in Cyber-Physical Systems: Foundations and Applications. Springer, pp. 105–112, https://doi.org/10.1007/978-3-030-67361-1

2. Dressler, F., & Tonguz, O. (2008). Self-Organization in Sensor and Actor Networks. Hoboken, NJ: Wiley, pp. 55–59 URL: https://www.wiley.com/en-us/Self-Organization+in+Sensor+and+Actor+Networks-p-9780470724453

3. Mahmoud, M. M., Gad, R. M., Younis, M., & Alghamdi, T. A. (2021). Secure Communication Protocols for Mobile Robotic Networks: Challenges and Solutions. *IEEE Communications Surveys & Tutorials*, *23*(2), pp. 1287–1312. https://doi.org/10.1109/COMST.2020.3048767

4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. Cambridge: MIT Press, pp. 305–317. URL: https://www.deeplearningbook.org

5. Delacour, K. J., & Wolf, C. (2020). A Survey of Neural Network Applications to Cryptography. *Neural Computing and Applications*, *32*(17), pp. 13349–13374. https://doi.org/10.1007/s00521-020-04859-9

Данчак О.
КНа-21, Національний університет "Львівська політехніка"
Науковий керівник к.т.н., доцент Вовк Олена Борисівна, кафедра СШІ
Національний університет "Львівська політехніка"

ПЕРЕОСМИСЛЕННЯ КОРПОРАТИВНОГО ІТ ЯК ЦЕНТРУ ІННОВАЦІЙ

Вступ. У XXI ст. інформаційні технології еволюціонували з допоміжного сервісу в ключовий елемент корпоративної стратегії. Компанії, що інтегрують ІТ у бізнес-процеси, демонструють на 30–45 % вищий прибуток від нових продуктів порівняно з конкурентами, які зберігають традиційну роль ІТ-функції. Актуальність теми обумовлена стрімким зростанням частки цифрових доходів у глобальному ВВП (20 % у 2023 р. за даними World Economic Forum) та необхідністю скорочення time-to-market під тиском динаміки споживчих очікувань. Саме тому розроблення методики перетворення корпоративного ІТ-підрозділу на центр інновацій є необхідною передумовою стійкого розвитку та збереження конкурентних позицій. Одним з ключових напрямів, що пришвидшує цю еволюцію, є автоматизований підбір кадрів із застосуванням моделей машинного навчання, який дозволяє оперативно формувати команди з необхідними компетенціями та мінімізувати ризики хибного найму.

Постановка задачі. Об'єктом дослідження є організаційні та технологічні чинники трансформації корпоративного ІТ. Предмет дослідження – автоматизація підбору кадрів методології, архітектурні патерни та процеси управління, що забезпечують прискорене перетворення бізнес-ідей у цифрові сервіси. Метою роботи є розроблення системної моделі, яка інтегрує продуктове мислення, сучасну архітектуру, управління даними та фінансові метрики в єдиний життєвий цикл створення цінності.

Основний матеріал. Культурна трансформація починається з переходу від проєктного до продуктового підходу, де принцип «you build it – you run it» покладає відповідальність за повний життєвий цикл на крос-функціональні squad-команди. Формування психологічно безпечного середовища й сервант-лідерства підвищує шанси досягнення високої продуктивності в 3,6 раза (State of DevOps 2024). Паралельно практика continuous discovery забезпечує безперервний зворотний зв'язок із замовниками, що скорочує цикл ухвалення рішень.

Технологічним фундаментом виступає мікросервісна та event-driven архітектура, реалізована через контейнеризацію та оркестрацію Kubernetes. Інфраструктура-як-код у поєднанні з GitOps гарантує відтворюваність середовищ та аудит змін, тоді як policy-as-code й service mesh підвищують керованість мережевих політик. Стратегія multi-cloud мінімізує залежність від постачальників і знижує ризики простою критичних сервісів.

Інформаційна компонента ґрунтується на концепції data fabric, що формує єдине джерело правди (SSOT) і спрощує інтеграцію аналітичних сервісів. Каталогізація метаданих і чіткі політики Data Governance гарантують якість даних, тоді як генеративний ШІ (LLM) автоматизує рутинні операції, підвищуючи точність прогнозів і персоналізацію клієнтського досвіду. За оцінкою Forrester (2024), впровадження GenAI скорочує time-to-market на 20 % і підвищує NPS на 10 пунктів.

Система метрик поєднує DORA-індикатори (deployment frequency, lead time for changes, MTTR) з OKR і FinOps-фреймворком, що дозволяє корелювати хмарні витрати з бізнес-результатами в реальному часі. Практика Lean Portfolio Management із пріоритизацією WSJF забезпечує баланс між операційними ініціативами та дисруптивними експериментами, тоді як shift-left-security та zero-trust-architecture знижують середній час відновлення сервісів до 30 хвилин. Комплексний ефект підтверджено кейсами рітейлу та фінтеху: перший скоротив цикл випуску маркетингових функцій із 28 днів до 72 годин, другий зменшив кредитний ризик на 9 %.

Окремо варто підкреслити роль ML-орієнтованого рекрутингу у формуванні високоефективних команд. Сучасні системи ATS, інтегровані з трансформерними NLP-моделями та механізмами Explainable AI, дозволяють за лічені хвилини провести попередній скринінг тисяч резюме, оцінити soft-skills кандидатів за відео-інтерв'ю та автоматично сформувати рейтинг кандидатів. Такі підходи скорочують середній час найму на

30–40 % та забезпечують вищу актуальність підібраних спеціалістів, що безпосередньо впливає на time-to-market IT-продуктів.

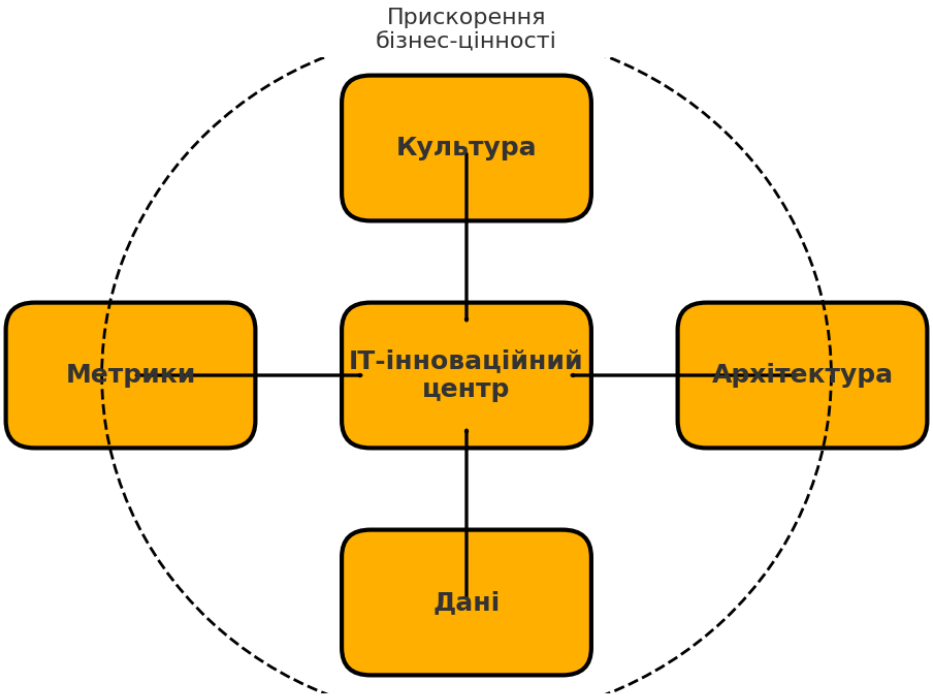


Рисунок 1 – Концептуальна модель IT-інноваційного центру

Таблиця 1 – Порівняння традиційної та інноваційної моделей IT

Характеристика	Традиційне IT	IT-центр інновацій	Ключові показники
Фокус	Підтримка операцій	Створення та масштабування цінності	Повернення інвестицій, приріст прибутку
Організація	Функціональні силоси	Крос-функціональні команди	Cycle time, lead time
Метрики	Uptime, SLA	OKR, DORA, NPS	Задоволення клієнтів, фінальних споживачів

Висновки. Перетворення корпоративного IT-підрозділу на центр інновацій є багатовимірним процесом, що охоплює культуру, архітектуру, дані та фінанси. Запропонована модель демонструє здатність скоротити time-to-market, підвищити задоволеність клієнтів і покращити фінансові показники, підтверджуючи доцільність і актуальність проведеного дослідження. Слід підкреслити, що автоматизований підбір кадрів, побудований на сучасних методах машинного навчання, є не лише допоміжним HR-процесом, а критичним механізмом перетворення корпоративного IT-підрозділу на справжній центр інновацій.

Список літератури

1. Gartner. (2024). Top Strategic Technology Trends 2024. Stamford, CT: Gartner Research.
2. McKinsey & Company. (2023). The IT Transformation Imperative. McKinsey Global Institute.
3. Forrester Research. (2024). The Total Economic Impact™ of Generative AI in the Enterprise.
4. World Economic Forum. (2023). Global Digital Economy Report.

Борисенко І.І.
ст. викладач кафедри кібербезпеки та програмного забезпечення ОНПУ

ГРАФОВА МОДЕЛЬ ЗЛОВМИСНИКА В ІНФОРМАЦІЙНІЙ СИСТЕМІ

Вступ. Наразі у науковому світі ведеться активна робота з математичного моделювання терористичних організацій та інших типів кримінальних груп. Традиційним шляхом для представлення групи людей з відображенням взаємних відносин між ними є використання теорії графів. Це зумовлено низкою факторів, серед яких наочність отриманої моделі, можливість адекватного відображення за допомогою стандартних операцій на графах реальних дій над групами та подій у групах, існування розробленого математичного апарату для роботи з графами, включаючи велику кількість, добре зарекомендувавших себе на практиці, евристичних методів обробки. У сучасному світі теорія графів є однією з найбільш актуальних
та найефективніших серед математичних технологій, бо сфера її застосування міститься у різних областях діяльності людства, зокрема у інформаційній та кібернетичній безпеці.

Аналіз наукової літератури свідчить про наявність зацікавленості вчених до проблеми використання графових технологій у кібербезпеці [1,2]:

- в інформаційній системі та у програмуванні;
- в моделюванні, аналізі та застосуванні графів атак;
-в криптографічних перетвореннях за допомогою теорії графів;
-у побудові дерева рішень у задачах прийняття рішень в умовах ризику і невизначеності.

Постановка задачі. Апріорне моделювання атак та їх наслідків охоплює і атаки і їх наслідки, але жодна з них не забезпечує оцінку впливу атаки при її незалежному використанні. Щоб позбутися цього недоліку, була побудована комплексна модель, яка дозволила точно оцінити вплив багатоступінчастості для виявлення шляхів відбиття атаки з високою ефективністю [3].

В більш ранніх роботах були розроблені моделі довільної групи зловмисника з строго обґрунтованим урахуванням ієрархії цієї групи за допомогою зваженого неорієнтованого графу. Вага вершин і ребер враховувала реальну цінність інформації, яка передавалась по лінії зв'язку.

Визначення вагових коефіцієнтів не є тривіальною задачею, навіть при наявності усієї необхідної інформації і ще більше ускладнюється у випадку, якщо таку інформацію треба зібрати.

Об'єктом дослідження є графові моделі в кібербезпеці. Предмет дослідження − застосування теорії графів до побудови графової моделі порушника в інформаційній системі.

Головна мета даного дослідження полягає в розробці графової моделі порушника, яка дозволить відмовитися від зваженого графу а також дозволить спростити її побудову та аналіз.

Основний матеріал. Будемо розглядати порушника інформаційної системи (ІС) як окремий випадок терористичної групи, а його математичну модель представимо орієнтованим графом. Окремі індивідууми в такій моделі представлені вершинами, пари яких з'єднуються орієнтованим ребром, якщо існує направлений зв'язок між ними, виходячи з апріорної інформації про порушника. Приклад графової моделі порушника представлений на рис.1.

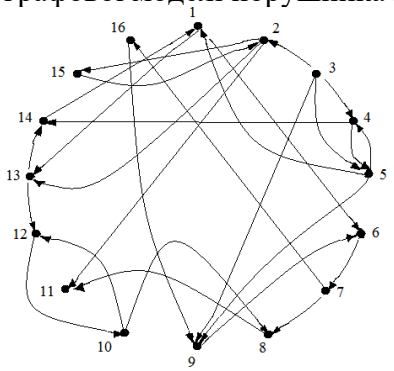


Рисунок 1 - Приклад графової моделі противника

Реальні моделі можуть мати достатньо складну структуру, в якій передбачити ієрархію вершин неможливо, тому перейдемо до більш зручного представлення графа.

Однією з основних задач, які розв'язуються по морфологічному аналізу орієнтованих графів, є розбивка множини його вершин на класи еквівалентності і порядку. Виконавши таку розбивку одержимо інше представлення графа, таке, як показано на рис. 2., з якого видно, що сильно зв'язані підграфи підпорядковані відношенню порядку. Такий граф легко піддається аналізу, оскільки впорядкування його класів еквівалентності очевидно, в реальних умовах це буває не так. Тому розглянемо простий алгоритм розкладу орграфа на класи порядка.

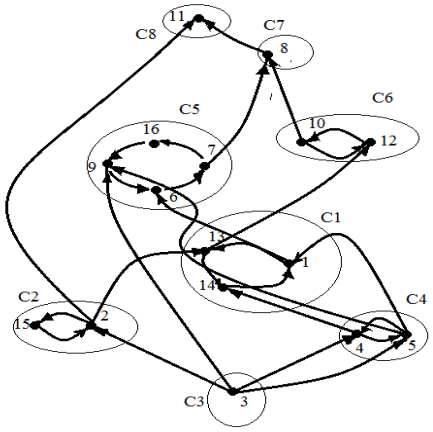


Рисунок 2. Графова модель зловмисника з виділеними сильно зв'язаними підграфами

Розбивці на класи порядку піддаються тільки орграфи, які не містять контурів і петель; якщо ж такі є, то спочатку треба виконати розбивку множини вершин на класи еквівалентності, а потім кожний з класів прийняти за вершину нового орграфа. В нашому випадку новий орграф буде мати вісім вершин C1, …, C8. Побудуємо його матрицю суміжності і запишемо її в масив Adj_factor. Масив Adj_factor має вигляд:

	1	2	3	4	5	6	7	8
1	0	0	0	0	1	1	0	0
2	1	0	0	0	0	0	0	1
3	0	1	0	1	1	0	0	0
4	1	0	0	0	1	0	0	0
5	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0

На кожному кроці алгоритму треба виявити стовпці, в яких відсутні одиниці. Такий стовпець на першому кроці алгоритму має номер 3 і відповідає вершині C3, яка представляє самий нижній рівень в ієрархії класів порядку, позначимо цей клас P1; у вершини C3 немає ні однієї дуги, що їй передує і тому вона є початковою (джерело).

Вилучаємо третій рядок і третій стовпець, які відповідають вершині C3, переходимо до масиву Adj1_factor. Знову відшукуємо стовпці, в яких відсутні одиниці – це стовпці з номерами 2 і 4, отже вершини C2 і C4 створюють клас P2 і т.д. В результаті одержимо шість класів порядку: P1 - C3; P2 – C2,C4; P3 - C1; P4 - C5,C6; P5 - C7; P6 – C8. Граф з розбивкою на класи порядку представлений рис. 3.

Треба зробити зауваження, що відразу треба починати розбивку графа на класи порядку, що запобігає зайвій роботі. Якщо на якомусь кроці не з'явився стовпець, який не містить одиниць, то це означало б, що вихідний граф містить контури і потрібна попередня розбивка його на класи еквівалентності.

Наприклад, якщо в нашому випадку ми відразу б почали виділяти класи порядку, пропускаючи крок розбивки на класи еквівалентності, то на першому кроці знайдеться стовпець, в якому немає одиниць – це стовпець під номером 3, але вже на другому кроці такого стовпця не одержимо.

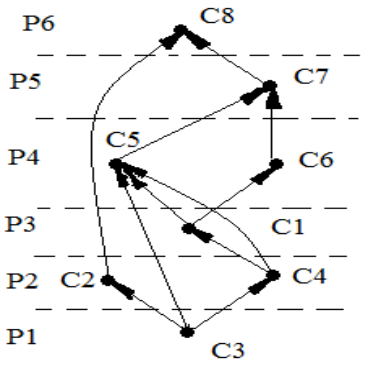


Рисунок 3 - Граф

Висновок. Традиційно графові моделі зловмисника використовуються для вирішення задач:

1. Визначення членів супротивника, блокування яких призведе до розпаду організації на декілька незв'язних між собою підгруп, а, отже, приведе як до повного припинення її функціювання так і до зменшення ефективності її діяльності.
2. Виділення в організації зловмисника таких зв'язків між її членами, вилучення яких приводить до розпаду групи на окремі частини, які не зв'язані між собою, що, очевидно, обмежить інформаційні можливості кримінальних структур.

В теорії графів перша задача відповідає задачі визначення мінімальної кількості вузлів, а друга – задачі визначення мінімальної кількості ребер, вилучення яких приводить вихідний зв'язний граф до незв'язного.

Проаналізуємо модель зловмисника (рис.3) з точки зору сказаного. Блокування підгруп С2 і С7, а також розірвання зв'язку (С1,С6) приводить до розпаду групи на 5 підгруп. Більш того, тільки одна підгрупа залишається зв'язана з джерелом – цей вузол слід трактувати як керівничу ланку. З іншого боку – вилучення зв'язків (С2,С8) і (С7,С8) приведе також до незв'язного графу, але чи призведе це до значних втрат, оскільки блоковані будуть тільки три члени групи.

Напроти, блокування тільки підгрупи С5 приведе до значних порушень інформаційних потоків і, очевидно, нанесе відчутну шкоду.

Виходячи зі сказаного можна зробити висновок, що хоч кількісну оцінку понесених втрат противника по одержаній графовій моделі зробити не можна, це не зменшує її переваг, оскільки запропонована модель відображує в повній мірі структуру передбачуваної організації і дає можливість планувати контрзасоби по послабленню її діяльності.

Не мало важливою перевагою є відмова від зваженого графу, що значно спростило побудову моделі, а представлення орієнтованого графу у вигляді класу еквівалентності і порядку привело до простого представлення складних і заплутаних вихідних графів.

Список літератури

1. Математичні моделі в кібербезпеці: графи та їх застосування в кібербезпеці: https://csecurity.kubg.edu.ua/index.php/journal/article/view/288/248
2. Cyber security: how to use graphs to do an attack analysis URL: https://linkurious.com/blog/cyber-security-use-graphs-attack-analysis/
3. Савченко В.А. Моделювання кібератак засобами теорії графів/ В.А. Савченко, О.Й. Мацько,С.В. Легомінова, І.С. Полторак, В.В. Марченко // *Сучасний захист інформації*, Київ, 2019. – №4(40). https://journals.dut.edu.ua/index.php/dataprotect/article/view/2351

Діденко А. Є.
аспірант 2 курсу кафедри ПЗ НУЗП
Науковий керівник д.т.н., професор Олійник А. О., кафедра ПЗ НУЗП

ПЕРЕВАГИ І НЕДОЛІКИ ВИКОРИСТАННЯ НЕЙРОМЕРЕЖЕВИХ МЕТОДІВ ДЛЯ ДІАГНОСТИКИ ТЕРМАЛЬНИХ ЗОБРАЖЕНЬ У СФЕРІ МЕДИЦИНИ

Вступ. Термографія – це процес отримання зображення теплового випромінювання за допомогою спеціальної камери (тепловізора). Завдяки своїх природі отримання даних, а саме неінвазивності, термографія знайшла широкий спектр використання в медичній сфері для детекції різноманітних патологій, таких як запальні процеси, рак, тощо. Для поліпшення аналізу отриманих зображень використовують нейромережеві методи комп'ютерного зору, такі як класифікація, детекція, сегментація, збільшення роздільної здатності.

Постановка задачі. Об'єкт дослідження – процес діагностики інфрачервоних медичних зображень. Предмет дослідження – нейромережеві методи глибинного навчання для аналізу медичних термальних зображень. Головна мета цього дослідження полягає в аналізі нейромережевих методів, які застосовуються в термографії, виділення основних напрямків і трендів, порівняння можливих переваг і недоліків для подальшої розробки власних нейромережевих архітектур.

Основний матеріал. Серед основних методів комп'ютерного зору є класифікація зображень (віднесення зображення до певного класу), детекція (виділення бажаних об'єктів на зображеннях), сегментація (розділення зображення на певні частини, відповідно до ознак кожного об'єкта), поліпшення якості зображення (знешумлення, збільшення роздільної здатності, видалення артефактів, тощо). Ці методи також можуть використовуватись для аналізу і обробки термальних зображень [1], шляхом створення і навчання відповідних нейромереж. Зазвичай, для цього використовують нейромережі, що були створені для аналізу звичайних RGB зображень (з подальшою модифікацією і покращенням), оскільки термальні зображення є їхнім піддоменом.

Класифікація медичних термальних зображень зазвичай зводиться до двох типів: бінарна класифікація, де позитивний клас – це наявність патології, а негативний – нормальний стан організма/органа, або багатокласова класифікація, де кожен клас представляє один із видів або ступенів певного захворювання. Серед поширених застосувань класифікації можна виділити аналіз термальних зображень за допомогою бінарної класифікації (здоровий орган або хворий орган), такі як діагностика раку грудей [2], захворювань печінки [3] виразка діабетичної стопи [4], тощо. Для цих цілей використовують популярні нейромережеві архітектури, такі як ResNet, InceptionNet, MobileNet, ViT, тощо. Такі моделі можуть досягати точності до 88% на відносно невеликих датасетах (до 1000 зображень) [1]. Це простий в імплементації підхід, але основним його недоліком є те, що він не показує, в якій саме області знаходиться хвороба.

Для більш точного аналізу використовують методи детекції і сегментації [5], які дозволяють локалізувати місце знаходження патології. Такі методи можуть бути як незалежними, так і бути в системі між собою (детекція з подальшою сегментацією) або з подальшою класифікацією [5]. Серед задач, що вирішуються цими методами, є детекція/сегментація раку грудей [5], пролежнів, тощо. Перевагою такого підходу є більш точна діагностика осередку хвороби, оскільки лікар має змогу побачити не тільки можливий клас хвороби, а і потенційну область на знімку. Однак, обсяги датасетів під детекцію або сегментацію досі залишаються малими, оскільки створення такої розмітки вимагає часу і експертності.

Збільшення роздільної здатності медичних зображень [6] дозволяє лікарям більш точно проаналізувати середовище захворювання, оскільки невеликі елементи (наприклад, вени, капіляри, контури органів, тощо) можуть бути частково втрачені під час отримання зображення. Наразі, дослідження методів ЗРЗ саме термальних медичних зображень майже відсутні через низьку кількість навчальних вибірок, але вони можуть бути запозичені (і в подальшому вдосконалені) із споріднених медичних сфер.

Незважаючи на широке застосування нейромереж для аналізу термальних зображень у сфері медицини, ці методи мають ряд недоліків, які необхідно враховувати при розробці системи комп'ютерного зору.

По-перше, це відносно малий об'єм навчальних вибірок. Невеликі набори даних зазвичай містять небагато зображень для кожного класу/об'єкту, що підвищує ризик перенавчання великих нейромереж [1]. Це змушує розробників використовувати невеликі моделі для аналізу зображень, що у свою чергу може знизити точність вихідного результату. Для вирішення цієї проблеми необхідно знайти оптимальний баланс між розміром навчальної вибірки і складністю моделі.

По-друге, термальні зображення у навчальних вибірках є досить специфічними і залежать від багатьох факторів, таких як якість камери, умови, в яких було зроблено знімок, фізичний стан пацієнтів, тощо. Це сильно впливає на репрезентативність кожної навчальної вибірки і робить створені моделі не універсальними. Для того, щоб зробити моделі більш генералізованими, необхідно, враховуючи вищезгадані фактори, використовувати та/або створювати репрезентативний набір даних (наприклад такий, що містить зображення з різних типів камер), застосовувати методи аугментації зображень.

Ще однією проблемою є інтерпретація та клінічна інтеграція створених моделей. Лікарі повинні мати змогу інтерпретувати вихідні результати та розуміти, яким чином було отримано той чи інший висновок. Звичайні нейронні мережі (особливо великі) вважаються "чорним ящиком", оскільки складно зрозуміти, як вони приймають рішення. Для вирішення цієї проблеми можна використовувати нейромережі, що базуються на механізмі уваги та/або продукують карту уваги (attention map) [7]. Такі карти можуть бути накладені на термограми, щоб виділити області, що впливають на діагноз.

Крім того, медичні датасети (не тільки термальні) часто мають високий дисбаланс між класами, де кількість "здорових" зображень в декілька разів перевищує кількість зображень із патологіями. Такий дисбаланс призводить до того, що моделям складно виявити ознаки патологій під час навчання. Рішенням проблеми може бути штучне розширення об'єму класу за допомогою аугментації, використання збалансованих функцій втрат під час навчання або регуляризації [8].

Висновки. Таким чином, способи застосування нейромереж для діагностики термальних зображень у сфері медицини показують багатообіцяючі результати. Враховуючи стрімкий розвиток нейромереж у сфері комп'ютерного зору, зокрема у медичній сфері, можна припустити, що спектр застосування термографії буде тільки розширюватись. Подальшими кроками дослідження можуть бути аналіз ресурсоефективних методів обробки медичних термальних зображень, аналіз методів збільшення об'єму набору даних та розробка власної архітектури нейромережі для діагностики медичних інфрачервоних зображень.

Список літератури

1. A. Z. Nowakowski, M. Kaczmarek, "Artificial Intelligence in IR Thermal Imaging and Sensing for Medical Applications". *Sensors*, 25(3):891, 2025.

2. A. Mashekova, Y. Zhao, E. Ng, V. Zarikas, S. C. Fok, O. Mukhmetov, "Early detection of the breast cancer using infrared technology – A comprehensive review". *Thermal Science and Engineering Progress* (27). 2025.

3. A. Özdil, B. Yilmaz. "Medical infrared thermal image based fatty liver classification using machine and deep learning". *Quantitative InfraRed Thermography Journal*, 21(2), 2023, pp. 102–119.

4. 6. K. Munadi., K, Saddami, M. Oktiana, et al., "A Deep Learning Method for Early Detection of Diabetic Foot Using Decision Fusion and Thermal Images". *Applied Sciences*, 12(15). 2022.

5. S. Civilibal, K. K. Cevik, A. Bozkurt, "A deep learning approach for automatic detection, segmentation and classification of breast lesions from thermal images". *Infrared Physics & Technology* (212), 2021.

6. H. Yang, Z. Wang, X. Liu et al., "Deep learning in medical image super resolution: a review". *Appl Intell*, 53, 2023, pp. 20891–20916.

7. M. Innat, M. F. Hossain, K. Mader, A. Z. Kouzani, "A convolutional attention mapping deep neural network for classification and localization of cardiomegaly on chest X-rays". *Scientific Reports*, 13, 2023.

8. A. Galdran, G. Carneiro and M. A. González Ballester, "Balanced-mixup for highly imbalanced medical image classification", *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference*, 2021, pp. 323-333.

Ватуляк О. В. (1), Рутецький Ю. О. (2)
1) магістрант НУ ЛП, 2) аспірант 3 курсу ФКІТ ЗУНУ,
Науковий керівник д.т.н., проф. Теслюк В.М., кафедра КІ ЗУНУ

РОЗРОБКА LSTM-МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ ЕНЕРГОСПОЖИВАННЯ У СМАРТ-БУДИНКАХ

Вступ. Проблема енергоефективності в умовах військової та енергетичної нестабільності є однією з ключових у сучасному суспільстві. Розумні будинки, оснащені системами моніторингу та керування енергоспоживанням, вимагають інструментів точного прогнозування енергопотреб. Одним з найперспективніших підходів є використання методів глибинного навчання, зокрема рекурентних нейронних мереж, таких як LSTM (Long Short-Term Memory), які дозволяють ефективно моделювати часові залежності у даних. Метою дослідження є розробка адаптивної LSTM-моделі для прогнозування споживання електроенергії у смарт-будинках з урахуванням як історичних спостережень, так і зовнішніх факторів (температура, час доби, день тижня тощо).

Постановка задачі. Актуальність розробки інтелектуальної системи прогнозування обумовлена потребою зменшення енергетичних втрат та оптимізації навантаження на енергомережі.

Об'єкт дослідження – процес споживання електроенергії у житлових приміщеннях.

Предмет – методи та алгоритми прогнозування на основі часових рядів.

Основні задачі дослідження:

- провести огляд існуючих методів прогнозування енергоспоживання;
- обґрунтувати доцільність використання LSTM-мереж;
- побудувати модель прогнозування;
- реалізувати програмний прототип та провести експерименти.

Основний матеріал. У сучасних дослідженнях широке застосування отримали методи машинного навчання для прогнозування енергоспоживання: лінійна регресія, дерева рішень, ансамблеві моделі (Random Forest), нейронні мережі. Ці моделі демонструють гарну продуктивність у статичних умовах, однак більшість з них не враховують часову динаміку даних, що суттєво знижує точність прогнозування у випадках нестаціонарного або сезонного споживання. Особливо важливим стає прогнозування у реальному часі, коли зміни у споживанні можуть бути спричинені як внутрішніми (поведінка мешканців), так і зовнішніми (температура, політична ситуація, аварії в мережі) факторами.

LSTM-моделі, як варіант рекурентної нейронної мережі, вирізняються здатністю зберігати довгострокову залежність між часовими точками та ефективно вирішують проблему «затухаючого градієнта». Вони були успішно застосовані в задачах прогнозування енергоспоживання, попиту на транспорт, медичних показників тощо. В роботах [1–3] наведено докази переваги LSTM у порівнянні з ARIMA, SVR, традиційними RNN у задачах з коротко- та довгостроковими залежностями.

У дослідженні було використано реальний датасет історичного енергоспоживання з розділенням на тренувальну, валідаційну та тестову вибірки [4]. Дані містили щогодинні показники споживання електроенергії протягом року. Попередня обробка включала очищення від пропущених значень, нормалізацію, виявлення та усунення викидів. Для генерації ознак були створені додаткові змінні: категорії часу доби, день тижня, середньодобова температура, свята. Було реалізовано стратегію ковзного вікна (sliding window), що дозволила формувати вхідні матриці для навчання мережі на основі заданої глибини історії.

Для моделювання використано бібліотеки TensorFlow і Keras. Підібрано оптимальну архітектуру LSTM: два послідовних шари LSTM з 64 та 32 нейронами відповідно, шар Dropout для запобігання перенавчанню, повнозв'язний шар з лінійною активацією на виході. Активно застосовувалися функції активації ReLU та оптимізатор Adam.

Для валідації точності були використані як класичні метрики похибки (MAE, MSE), так і коефіцієнт детермінації , що показав, наскільки модель наближається до ідеального прогнозу.

Перехресна валідація допомогла уникнути переоцінки моделі на обмеженому обсязі даних. Також аналізувалася стабільність моделі на нових (небачених) відрізках часу, зокрема у кризових зимових періодах.

Архітектура реалізована у вигляді модульної системи, що використовує Docker-контейнери для ізоляції компонентів. Основні модулі:

- обробка та зберігання даних (ETL);
- модуль підготовки навчальних даних;
- тренувальний модуль з LSTM;
- API для прогнозування у режимі реального часу;
- візуалізація результатів у форматі дашборду.

На рисунку 1 представлена структура LSTM-моделі, яка включає:

- Input Layer: 24 часові кроки
- LSTM Layer 1: 64 нейрони з return_sequences=True
- LSTM Layer 2: 32 нейрони
- Dropout Layer: для запобігання перенавчанню
- Dense Layer: вихід одинарного значення прогнозу

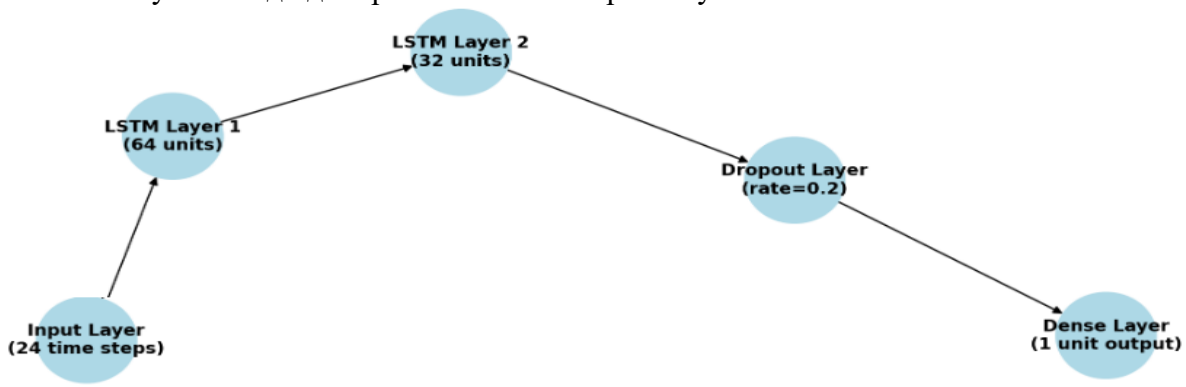


Рисунок 1 – Архітектура LSTM-моделі

Результати експериментів. Проведено серію експериментів з використанням різних гіперпараметрів. Найкращі результати досягнуті при глибині мережі 2 шари та розмірі вікна 24. MAE склало 0,13, а MSE – 0,025. Прогнозна модель показала стабільну поведінку на тестовій вибірці з високим ступенем відповідності реальним спостереженням. Запропонована модель також дозволяє адаптуватися до змін у шаблонах споживання, що є критично важливим у кризових умовах.

Висновки. Розроблено інтелектуальну систему прогнозування енергоспоживання на основі LSTM-моделі. Дослідження підтвердило її ефективність у реальних умовах. Результати експериментів свідчать про високу точність та адаптивність моделі. Отримані результати можуть бути використані у смарт-системах для оптимізації енергоресурсів, побудови рекомендацій та реагування на пікові навантаження. У перспективі – розширення функціональності системи, включення компонентів самонавчання та інтеграція з IoT-пристроями.

Список літератури

1. Maarif, M. R., Saleh, A. R., Habibi, M., Fitriyani, N. L., & Syafrudin, M. (2023). Energy Usage Forecasting Model Based on Long Short-Term Memory (LSTM) and eXplainable Artificial Intelligence (XAI). Information, 14(5), 265. https://doi.org/10.3390/info14050265

2. Alghamdi, M. A., AL–Malaise AL–Ghamdi, A. S., & Ragab, M. (2024). Predicting energy consumption using stacked LSTM snapshot ensemble. Big Data Mining and Analytics, 7(2), 247–270. https://doi.org/10.26599/BDMA.2023.9020030

3. Yan, K., Li, W., Ji, Z., Qi, M., & Du, Y. (2019). A hybrid LSTM neural network for energy consumption forecasting of individual households. IEEE Access, 7, 157633–157642. https://doi.org/10.1109/ACCESS.2019.2949065

4. GitHub. (2024). Dataset: Household power consumption. https://media.githubusercontent.com/media/dmkn96/Power_consumption/refs/heads/main/Dataset/household_power_consumption.txt

SERGII KUZMIN
PhD Student at Department of Automated Control Systems, Lviv Polytechnic National University
Academic Supervisor Dr. Sci. (Eng), Professor OLEH BEREZSKY, Head of Computer Engineering Department, West Ukrainian National University

EFFICIENT FINE-TUNING OF STABLE DIFFUSION FOR BIOMEDICAL IMAGE GENERATION

Introduction. The past decade has witnessed a rapid shift from *generative adversarial networks* to *denoising diffusion probabilistic models* (DDPMs) for high-fidelity image synthesis. Diffusion models learn to reverse a gradual noising process and thereby avoid the mode-collapse and training-instability pitfalls typical of GANs, achieving more stable convergence and broader coverage of complex data distributions. Their strength in reproducing fine structural detail makes them an especially compelling choice for biomedical domains - histopathology, radiology, and microscopy - where subtle texture cues can be diagnostically decisive.

Early biomedical studies confirm this potential. For example, Niehues *et al.* fine-tuned a latent diffusion model on colon slides and showed that synthetic patches improved downstream cancer-classification accuracy by four percentage points when used for data augmentation [1]. Such results highlight two intertwined realities:

1. *Domain shift.* A diffusion model pretrained on generic Internet imagery must be adapted to domain-specific staining patterns and cellular morphologies before its outputs can be trusted in clinical pipelines.

2. *Resource limits.* Medical datasets are small and privacy-sensitive, and many laboratories operate on a single 16 GB GPU, making full-parameter retraining impractical.

To bridge this gap, the community has proposed *parameter-efficient adaptation* techniques. Two of the most widely adopted, yet methodologically contrasting, approaches are:

- *DreamBooth* - full-model fine-tuning with a class-specific prior-preservation loss, delivering strong fidelity at the cost of updating all ≈ 860 M weights of Stable Diffusion;
- *HyperNetwork* - a lightweight auxiliary network that dynamically modulates only selected backbone layers, trimming learnable parameters by orders of magnitude and enabling adaptation on mid-range GPUs.

Although both methods have been validated individually [2, 3], the literature still lacks a focused, quantitative comparison of their trade-offs in biomedical image synthesis. Consequently, this thesis covers an in-depth, head-to-head study of DreamBooth versus HyperNetwork for histopathology image generation. By concentrating on the two most contrasting and prevalent paradigms, the research provides:

- a unified evaluation protocol using standard generative metrics;
- evidence-based guidance for practitioners who must balance image realism against hardware and privacy constraints.

These contributions are intended both to satisfy an immediate clinical need for trustworthy synthetic data and to lay the methodological foundation for a forthcoming broader study that will encompass additional fine-tuning strategies.

Problem Statement. Despite the promise of diffusion models for biomedical imaging, laboratories still lack evidence on whether *DreamBooth* or *HyperNetwork* offers the superior quality-versus-compute trade-off under scarce data, privacy constraints, and modest GPUs. The object of this study is the adaptation of large diffusion models to specialised biomedical domains; the subject is the head-to-head use of *DreamBooth* and *HyperNetwork* for histopathology synthesis; the goal is to pinpoint which method best balances fidelity, diversity, and resource efficiency, thereby supplying practical guidance to resource-constrained biomedical-AI teams.

Main Material. *DreamBooth* introduces a novel visual concept into a pretrained text-to-image diffusion pipeline by binding that concept to a rare identifier token and then updating *all* model weights so that the token unambiguously evokes the new subject. Training is organized in two sequential stages:

1. *Base-model fine-tuning (64 × 64 px).* The U-Net and cross-attention layers are trained on 64 px crops with prompts like "a photo of <*S*> tissue," while a parallel class-preservation loss on generic prompts ("a photo of a tissue") keeps class diversity intact as the model learns the new subject.

2. *Super-resolution fine-tuning (64 → 512 px)*. Two stacked up-sampling U-Nets train on paired low-/high-resolution crops of each tissue, injecting micro-architectural and staining detail into the final 512 px images.

Because every parameter of the diffusion backbone and super-resolution modules is updated (≈ 860 M weights in Stable Diffusion 1.5), *DreamBooth* achieves high prompt fidelity and visual accuracy at the cost of increased VRAM usage and longer optimization time relative to adapter-based techniques.

HyperNetwork adapts a large text-to-image diffusion model without touching its original weights. Instead of re-optimizing the entire backbone, the method adds a lightweight auxiliary network H that produces task-specific weight offsets for a selected subset of layers in the base model F.

Formally, let θ be the frozen parameters of F and φ the learnable parameters of H. Given a conditioning vector c - typically a learned embedding that encodes the target concept - H outputs a modulation tensor $\Delta\theta = H(c; \varphi)$. At run-time the effective weights of the chosen layers are

$$\widehat{\theta} = \theta + \Delta\theta, \tag{1}$$

so the diffusion process uses $\widehat{\theta}$ while the original θ remains intact. Only φ (often < 1 % of the 860 M parameters in Stable Diffusion) is updated during training.

Training proceeds with the same noisy-denoising objective as the underlying diffusion model: given an input prompt p (with a placeholder token) and a reference biomedical image x, the optimization minimizes a reconstruction / diffusion loss

$$L_{diff}\left(x, F_{\widehat{\theta}}(p)\right), \tag{2}$$

driving H to generate modulations that reproduce the desired tissue appearance, nuclear morphology, and staining characteristics.

Because the bulk of Stable Diffusion stays frozen, adaptation fits comfortably into 16 GB of VRAM and completes in a fraction of the time required for full-model fine-tuning, making *HyperNetwork* an attractive option for resource-constrained biomedical labs.

Fine-tuning experiments were conducted on Stable Diffusion 1.5 model using publicly available Chaoyang dataset, which contains colon slides from Chaoyang hospital with 512 × 512 patch size [4]. For the experiments the dataset was rebalanced resulting in the same number of images across different tissue types for both training and testing sets. 664 samples of each type for the training set and 273 samples of each type for the testing set were randomly selected from the base dataset.

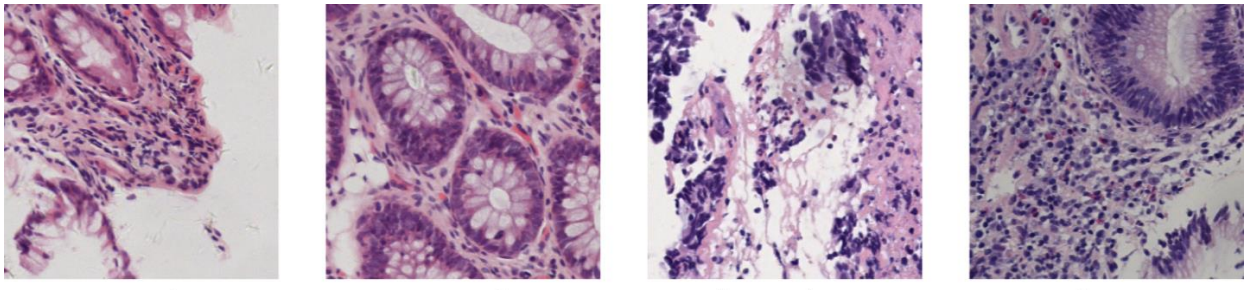


Figure 1. Histological samples from Chaoyang dataset [4].

The training was carried out utilizing the base Stable Diffusion 1.5 model (stabilityai/stable-diffusion-v1-5) on the local environment with Windows 11 Pro 64-bit operating system and the following hardware, which reflects a configuration with limited VRAM at disposal: Intel Core i9-13980HX, 32 GB DDR 5 RAM, and NVIDIA GeForce RTX 4090 Laptop GPU with 16 GB of GDDR6 VRAM.

DreamBooth trainings were conducted using Kohya_SS v24.1.7. *HyperNetwork* training and image generation with both fine-tuned models were done using Stable Diffusion web UI v1.10.1 (AUTOMATIC1111). The training parameters for each fine-tuning method are presented below.

Table 1 Training parameters for DreamBooth and HyperNetwork fine-tuning

DreamBooth				**HyperNetwork**	
Repeat Factor	2	Unet Learning Rate	1e-5	Batch Size	4
Batch Size	2	Optimizer	AdamW 8-bit	Base Learning Rate	5e-5
Epoch	12	Seed	22	Layer Structure	[1, 2, 4, 2, 1]
Mixed Precision	fp16			Activation Function	Linear
Warmup steps	5%			Training Steps	4500
Text Encoder Learning Rate	5e-5				

Inference used DPM++ 2M with a Karras scheduler, 50 sampling steps, CFG 7.5, and 512 × 512 resolution. To quantitatively assess the quality of the synthesized images, three widely used metrics in generative model evaluation: Fréchet Inception Distance (FID), Precision, and Recall were employed. These metrics evaluate realism, fidelity, and diversity of generated samples by comparing them to real data in a learned feature space.

Table 2. FID, Precision, and Recall metrics for images of all tissue types generated with DreamBooth and HyperNetwork.

	normal			**serrated**			**adenocarcinoma**			**Adenoma**		
	FID	Precision	Recall	FID	Precision	Recall	FID	Precision	Recall	FID	Precision	Recall
DreamBooth	**97,12**	**0,3443**	0,6850	127,06	0,2308	**0,7766**	122,14	**0,5531**	0,5128	121,61	0,1245	**0,7619**
HyperNetwork	101,22	0,1392	**0,7253**	**107,59**	**0,4579**	0,5934	**77,27**	0,3883	**0,5678**	**87,34**	**0,3223**	0,6557

Representative qualitative results are shown in Figure 2.

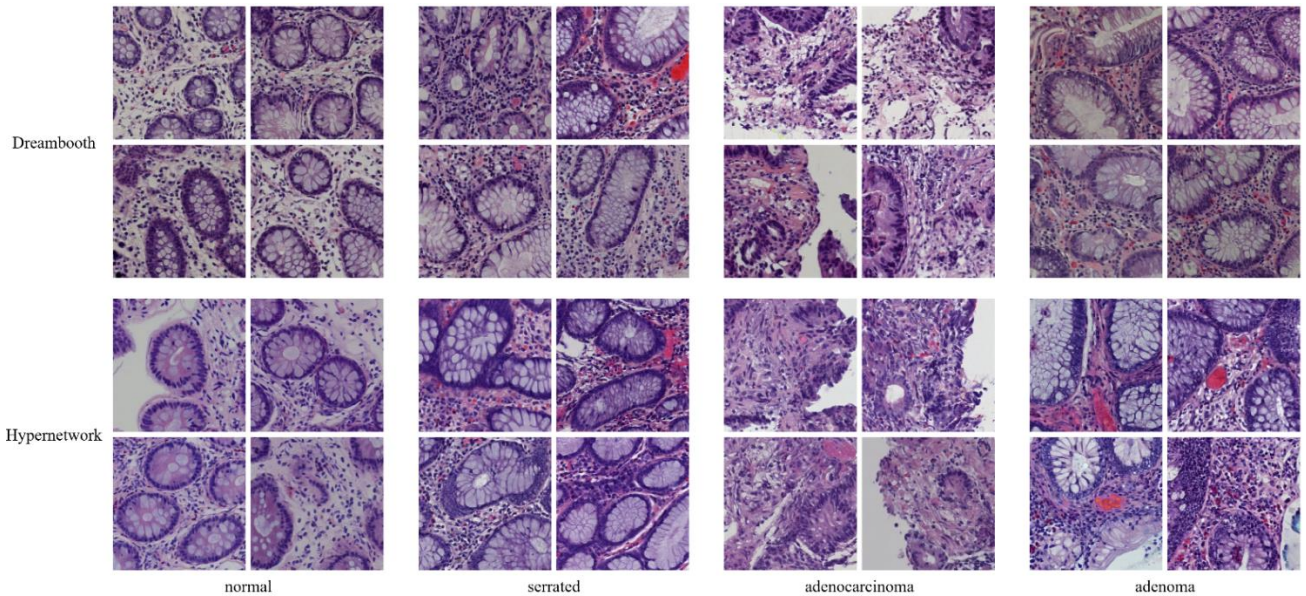


Figure 2. Selection of DreamBooth- and Hypernetwork-generated patches.

Conclusions. *HyperNetwork* proved able to match - or even surpass - *DreamBooth* on three of four colon-tissue classes, achieving comparable fidelity while updating less than 1 % of Stable Diffusion's weights. *DreamBooth* retained an edge only for routine "normal" tissue but did so at the cost of re-optimizing the full 860 M-parameter backbone and consuming markedly more VRAM. These findings confirm that adapter-based tuning and full-model fine-tuning occupy different points on a

precision-versus-compute spectrum: HyperNetwork favours resource-limited, privacy-constrained settings and data-augmentation pipelines, whereas DreamBooth remains preferable when absolute visual faithfulness is paramount and hardware budgets are ample. The study constitutes the first controlled head-to-head comparison demonstrating that a lightweight adapter can outperform full-model adaptation in medical-image diffusion, supplying biomedical-AI teams with clear, evidence-based guidance for balancing generative quality against real-world deployment constraints.

References

1. Niehues, J. M., et al. (2024). Using histopathology latent diffusion models as privacy preserving dataset augmenters improves downstream classification performance. Computers in Biology and Medicine. https://doi.org/10.1016/j.compbiomed.2024.108410
2. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Aberman, K. (2022). DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. arXiv preprint arXiv:2208.12242. https://doi.org/10.48550/arXiv.2208.12242
3. Ruiz, N., et al. (2023). HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models. arXiv preprint arXiv:2307.06949. https://arxiv.org/abs/2307.06949
4. Zhu, C., Chen, W., Peng, T., Wang, Y., & Jin, M. (2022). Hard Sample Aware Noise Robust Learning for Histopathology Image Classification. IEEE Transactions on Medical Imaging. https://doi.org/10.1109/TMI.2021.3125459

Лящинський П.Б.
аспірант 3 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Березький О.М., кафедра КІ ЗУНУ

ГЛИБОКЕ НАВЧАННЯ В МЕДИЧНІЙ ДІАГНОСТИЦІ

Вступ. У сучасній медичній практиці все більшої важливості набувають методи автоматичного аналізу медичних зображень. Глибоке навчання, зокрема згорткові нейронні мережі (ЗНМ), відкривають нові можливості для точного та швидкого розпізнавання патологій.

Наукова група під керівництвом професора Олега Березького, яка працює у Західноукраїнському *національному* університеті, на протязі двадцяти років займається застосуванням штучного інтелекту в системах комп'ютерної діагностики. У ряді наукових робіт групи дослідників відображені методи, алгоритми та програмні засоби для діагностування в онкології [1-6].

Постановка задачі. Об'єкт дослідження – процеси опрацювання, зберігання, сегментації та класифікації біомедичних зображень на основі згоркових нейронних мереж. Предмет дослідження – методи, моделі та засоби опрацювання, зберігання, сегментації та класифікації біомедичних зображень на основі згорткових нейронних мереж. Мета дослідження – аналіз методів глибокого навчання, що застосовуються для автоматичної сегментації та класифікації біомедичних зображень, з оцінкою їхнього потенціалу в медичній діагностиці.

Основний матеріал. Для постановки діагнозу раку та підтипів раку молочної залози використовують гістопатологічні та імуногістохімічні зображення.

Приклад гістопатологічних зображень для класів G1, G2 та G3 приведено на рисунку 1.

Гістопатологічні зображення дозволяють ідентифікувати рак молочної залози.

Імуногістохімічне дослідження використовує особливі антитіла для виявлення білків чи антигенів у клітинах тканин. Діагностуючий зразок обробляється антитілами, що зв'язуються з певними білками, які є важливими для ідентифікації типу клітин чи патологічного процесу.

Виділяють чотири основних молекулярно-генетичних підтипи раку молочної залози: люмінальний A; люмінальний B; HER-2/neu-ампліфікований; базальноподібний.

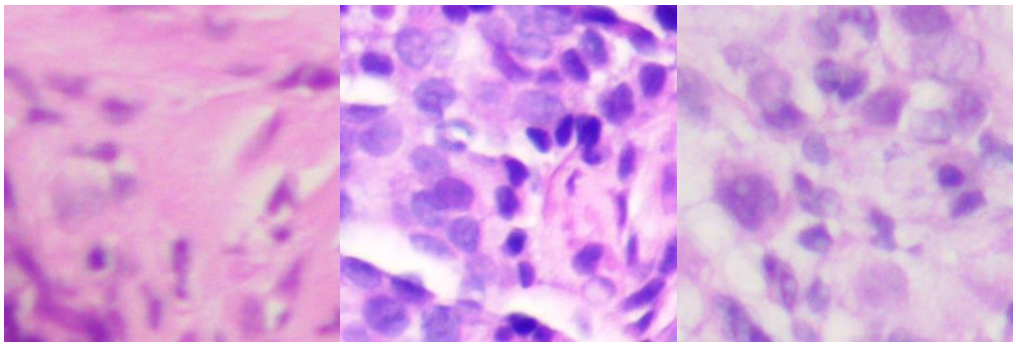


Рисунок 1 – Гістопатологічні зображення

Отже, для одного гістопатологічного зображення є чотири імуногістологічних на основі яких визначають підтип раку молочної залози. При цьому використовують два інформативних параметри: відносну площу клітин, які зреагували на біомаркер і середню інтенсивність цих клітин.

Приклад оригінального та сегментованого імуногістохімічного зображення приведено на рисунку 2.

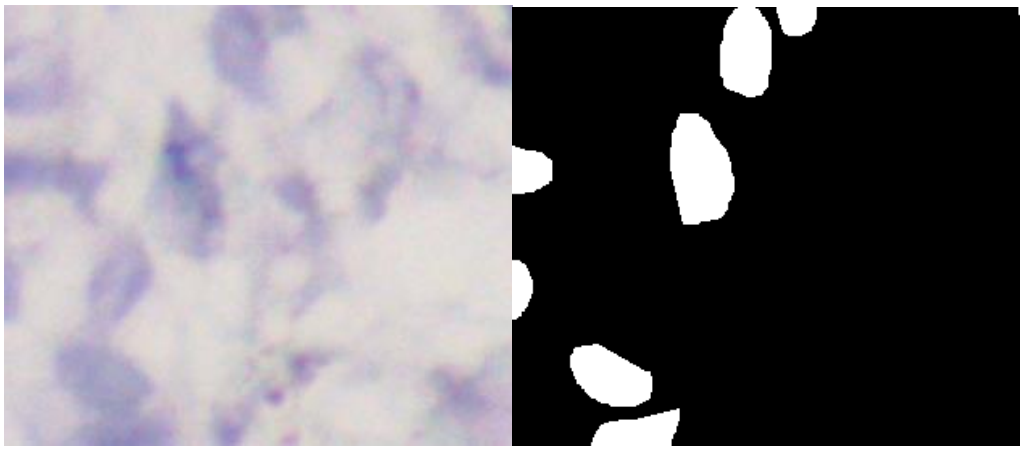


Рисунок 2 – Імуногістохімічне зображення (оригінальне та сегментоване)

Згорткові нейронні мережі (зокрема мережі типу U-Net) широко використовуються в задачах сегментації для автоматичного виділення клітин або структур, що зреагували на біомаркери. Це, в свою чергу, дозволяє обчислювати кількісні параметри, зокрема відносну площу реактивних клітин та середню інтенсивність їх фарбування.

Окрім сегментації, ЗНМ активно застосовуються і в задачах класифікації гістопатологічних зображень. Вони дозволяють автоматично визначати наявність патологічних змін та ідентифікувати підтипи пухлин на основі морфологічних та кольорових ознак тканин. Завдяки здатності виявляти складні шаблони візуальних даних, згорткові нейронні мережі можуть навчатися розрізняти підтипи раку молочної залози, використовуючи комбінацію кількісних характеристик, таких як інтенсивність забарвлення та просторове розташування клітин, що зреагували на відповідні біомаркери.

Автоматичне діагностування має великий потенціал, але стикається з певними викликами [3]:

1. *Точність і надійність:* потрібно забезпечити високу точність діагнозів, щоб уникнути помилкових висновків.
2. *Етика і конфіденційність:* забезпечення захисту персональних даних пацієнтів.
3. *Інтерпретація результатів:* надання інтерпретованих результатів для лікарів та пацієнтів.

У роботі [3] було розроблено метод автоматичного діагностування на основі нейронних мереж. Його основна суть полягає в паралельній сегментації та класифікації біомедичних зображень із подальшим комбінуванням результатів у єдиний формалізований звіт.

Якісне медичне діагностування значною мірою залежить від оптимізації архітектури нейронних мереж, оскільки наявні підходи часто не враховують структурну специфіку тканин різних типів, що, у свою чергу, знижує точність моделей у реальних клінічних умовах.

Процес пошуку архітектури містить такі кроки: ініціалізація, пошук гіперпараметрів, тренування, оцінка та оптимізація.

На першому кроці задаються базові параметри моделі, такі як розмір початкового фільтра, кількість шарів та швидкість навчання. Крім того, можна задати послідовність операцій чи блоків моделі.

Пошук гіперпараметрів включає автоматичне тестування різних варіантів параметрів, що задаються в *search_space* конфігурації засобами Neural Network Intelligence (NNI) для знаходження найкращого варіанту. Далі модель тренується із заданими параметрами. При цьому, на кожному кроці коригуються ваги за допомогою зворотного поширення помилки. Після кожної епохи обчислюються метрики, що дає змогу визначити ефективність моделі на валідаційних даних.

На останньому етапі NNI обирає найкращі параметри на основі заданих метрик точності. Після цього все знову починається з пошуку гіперпараметрів. Це дозволяє покращити ефективність моделі на наступних етапах. Так триває доти, поки не буде знайдено задану точність чи не вичерпаються ресурси для пошуку оптимальної архітектури.

Висновки. Сегментація зображень дозволяє виділяти області клітин, які зреагували на відповідні біомаркери, що забезпечує отримання кількісних показників (площа, інтенсивність), критично важливих для оцінки підтипу пухлини.

Класифікаційні моделі на основі ЗНМ здатні розрізняти підтипи пухлин за морфологічними та колірними характеристиками, що дозволяє значно підвищити точність діагнозу та автоматизувати процес інтерпретації результатів.

Оптимізація архітектур нейронних мереж**,** зокрема за допомогою інструментів автоматичного пошуку, таких як NNI, дозволяє адаптувати моделі до специфіки біомедичних даних і підвищити їхню ефективність у клінічних умовах.

Інтеграція методів сегментації та класифікації в єдину систему автоматичного аналізу створює підґрунтя для розробки повноцінних рішень у сфері комп’ютерної медицини з потенціалом застосування в реальній клінічній практиці.

Список літератури

1. Berezsky, O.M., Liashchynskyi, P.B. (2021). *Comparison of generative adversarial networks architectures for biomedical images synthesis. Applied Aspects of Information Technology*, 4(3), 250–260. https://doi.org/10.15276/aait.03.2021.4

2. Berezsky, O.M., Liashchynskyi, P.B. (2025). *Development of the architecture of a computer aided diagnosis system in medicine. Applied Aspects of Information Technology*, 7(4), 359–369. https://doi.org/10.15276/aait.07.2024.25

3. Berezsky O.M., Liashchynskyi P.B., Pitsun O., Izonin I. (2024). *Synthesis of Convolutional Neural Network architectures for biomedical image classification. Biomedical Signal Processing and Control*, 95, 106325.

4. Berezsky, O., Liashchynskyi, P., Melnyk, G., Dombrovskyi, M., & Berezkyi, M. (2024, November). *Synthesis of biomedical images based on generative intelligence tools. In 7th International Conference on Informatics & Data-Driven Medicine*. University of Birmingham, Birmingham, United Kingdom.

5. Berezsky, O.M., & Liashchynskyi, P.B. (2024). *METHOD OF GENERATIVE-ADVERSARIAL NETWORKS SEARCHING ARCHITECTURES FOR BIOMEDICAL IMAGES SYNTHESIS. Radio Electronics, Computer Science, Control*, (1), 104–104.

6. Oleh, B., Oleh, P., Melnyk, G., Datsko, T., Izonin, I., & Derysh, B. (2024). *An Approach toward Automatic Specifics Diagnosis of Breast Cancer Based on an Immunohistochemical Image. Journal of Imaging*, 9.

7. Лящинський, П. (2023). *Інтеграція штучного інтелекту в медичній діагностиці*. // *VIII Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі»*, 5 грудня 2023 р.: тези доп. – Тернопіль, 2023. – С. 35.

8. Лящинський, П. (2024). *Аналіз медичних систем комп'ютерної діагностики*. // *IX Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі»*, 21 травня 2024 р.: тези доп. – Тернопіль, 2024. – С. 36–37.

9. Лящинський, П. (2024). *Архітектура системи комп'ютерного діагностування*. // *X Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі»*, 5 листопада 2024 р.: тези доп. – Тернопіль, 2024. – С. 32–33.

10. Лящинський, П. Б. (2024). *Програмний засіб для класифікації та синтезу біомедичних зображень. Scientific Bulletin of UNFU*, 34(4), 120–127.

11. Лящинський, П., & Лящинський, П. (2023). *ANALYSIS OF METRICS FOR GAN EVALUATION*. *Computer systems and information technologies*, (4), 44–51.

12. Лящинський, П.Б., Березький, О.М. (2025). *Системи комп'ютерного діагностування: методи та засоби*. *Український журнал інформаційних технологій*, 6(2), 57–63. https://doi.org/10.23939/ujit2024.02.057

Прокопів А.Р.
аспірант 1 курсу ІКНІ НУ «Львівська Політехніка»
Науковий керівник д.т.н. Медиковський М.О., ІКНІ НУ «Львівська Політехніка»

МЕТОД АНАЛІЗУ ВЛАСНИХ ЧАСТОТ ДЛЯ ВИЗНАЧЕННЯ ПОШКОДЖЕНЬ ЛОПАТЕЙ ВІТРОВОЇ ЕЛЕКТРИЧНОЇ УСТАНОВКИ

Вступ. Сучасна енергетика активно інтегрує відновлювані джерела енергії, зокрема вітрові електричні установки, які забезпечують екологічно чисту електроенергію. Надійність, ефективність та безпека їх експлуатації безпосередньо залежать від технічного стану складових елементів, особливо лопатей вітрових установок. Дощ, град, блискавки, обмерзання, удари об поверхню лопаті на великій швидкості – все призводить до утворення дефектів, таких як ерозія, тріщини, розшарування та інші. Для зниження витрат на технічне обслуговування та скорочення періодів простою необхідно використовувати сучасні інформаційні технології моніторингу.

Постановка задачі. Об'єкт дослідження – процес визначення пошкоджених лопатей вітрової електричної установки (ВЕУ). Предмет дослідження – методи операційного модального аналізу (ОМА) для класифікації лопатей ВЕУ. Мета дослідження – підвищити ефективність та визначити можливості застосування ОМА для визначення пошкоджених лопатей ВЕУ.

Основний матеріал. У цьому дослідженні зроблено фокус на неруйнівному контролі (НК) стану лопатей ВЕУ. В Державному Стандарті України виділено такі види НК: магнітний, електричний, електромагнітний, радіохвильовий, оптичний, тепловий, радіаційний, акустичний, органолептичний, візуальний, електрогазодинамічний та метод проникних речовин. Однак, кожен з них має як свої переваги, так і недоліки.

Одним з найперспективніших методів НК є аналіз власних частот, оскільки він дозволяє дистанційно виявляти зміни конструкції за допомогою сенсорів вібрації, акселерометрів або тензометрів встановлених на лопаті [1]. Коливальні системи мають тенденцію коливатися синусоїдально з однаковою частотою за відсутності збурень. Це називають власною частотою. Власні частоти конструкцій визначаються їх нелінійними масовими та жорсткісними характеристиками. Поява дефектів, таких як тріщини, локальні ослаблення або розшарування, призводить до зниження жорсткості та змінює спектр власних частот. Дослідження показали що зміни власних частот є істотними і можуть бути використаними для виявлення пошкоджень. Цей вплив є більш явний на більшій швидкості обертання ВЕУ. Загалом методи аналізу власних частот передбачають виконання таких етапів: визначення власних частот непошкодженої лопаті (еталон); збір даних для визначення власних частот під час експлуатації; порівняння отриманих значень з еталонними; аналіз відхилень для встановлення факту пошкодження.

Методи Операційного Модального Аналізу (OMA) дозволяє визначати модальні параметри зі сигналу вібраційної відповіді за допомогою автокореляційних та крос-кореляційних функцій [2]. Основна відмінність порівняно з традиційним експериментальним модальним аналізом полягає в тому, що немає необхідності вимірювати вхідні збудження. При цьому система має відповідати трьом основним припущенням. Вона повинна бути лінійною та незмінною в часі (Linear Time Invariant), збуджувальні сили мають бути представлені спектром білого шуму з рівномірною щільністю потужності в робочому діапазоні частот, а також вони повинні бути некорельовані. Чим краще виконуються ці припущення, тим вищою буде якість оцінених модальних параметрів [3]. Використано реалізацію OMA яка базується на методі вибору піків спектральної щільності. Він передбачає такі етапи:

1. Вимірювання сигналів (наприклад прискорення з акселерометрів);
2. Обчислення спектральної щільності потужності. Це отримується з квадрату перетворення Фур'є. Перетворення Фур'є визначається залежністю

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{kn}{N}},$$

де X(k) – k-ий компонент перетворення Фур'є, k – індекс в частотній області, x(n) – n-ий компонент вхідної послідовності, n – індекс в часовій області, N – кількість відліків вхідної послідовності.

3. Визначення частот піків спектральної щільності потужності.

Частоти, на яких виникають ці піки, є гармоніками власної частоти. Проте цей метод має недоліки – він є чутливим до шуму та не може розрізнити близько розташовані моди. Для того щоб подолати ці недоліки, використовується метод Стохастичної Ідентифікації Підпростору (SSI) [4]. Термін *підпростір* означає, що метод базується на ідентифікації моделі у просторі станів та включає етап усічення шляхом сингулярного розкладу (SVD).

У цьому методі так звана стохастична модель у просторі станів визначається за кореляціями вихідних сигналів або безпосередньо з виміряних вихідних даних. Модель у просторі станів записується у вигляді рівняння:

$$\begin{cases} x_{k+1} = \boldsymbol{A} x_k + w_k \\ y_k = \boldsymbol{C} x_k + v_k \end{cases},$$

де y_k — вектор дискретних вихідних даних, x_k — вектор станів системи, w_k — процесний шум, спричинений збуреннями та невідомими збудженнями конструкції, v_k — шум вимірювання, переважно через похибки датчиків, k — дискретний момент часу. Матриця **A** є матрицею переходу станів, що повністю описує динаміку системи через власні значення; **C** — матриця виходів. Єдиними відомими величинами в рівнянні є вихідні вимірювання y_k, а основне завдання полягає у визначенні системних матриць **A** та **C**, на основі яких можна обчислити модальні параметри. Обчислення модальних параметрів починається з розкладу матриці **A** за власними значеннями:

$$\boldsymbol{A} = \boldsymbol{\Psi} \boldsymbol{\Lambda}_d \boldsymbol{\Psi}^{-1},$$

де $\boldsymbol{\Psi}$ – матриця власних векторів, $\boldsymbol{\Lambda}_d$ - діагональна матриця, що містить дискретні власні значення μ_i, які пов'язані з безперервними власними значеннями λ_i за співвідношенням: $\mu_i = \exp(\lambda_i \Delta t)$. Власні частоти та коефіцієнти загасання визначаються через λ_i за відношенням:

$$\lambda_i \lambda_i^* = -\xi_i \omega_i \pm j\sqrt{1 - \xi_i^2}\,\omega_i,$$

де ω_i – власні частоти.

Даний метод має перевагу в тому, що уточнює оцінки власних частот та є більш ефективним в середовищах з шумом. Проте цей метод є більш обчислювально інтенсивним та чутливим до розміру вектору станів системи (x_k).

В ході дослідження виконано OMA на існуючих та симульованих даних за допомогою програми OpenFAST. Найбільшою була різниця першої гармоніки, та при порівнянні лопаті з тріщиною та лопаті без пошкоджень було виявлено відхилення -1.4%. Різниця власних частот лопаті з ерозією та лопаті без пошкоджень була меншою та складала -0.6%.

Висновки. З використанням розробленої інформаційної технології підтверджено можливості використання OMA для визначення пошкодження лопаті ВЕУ. Особливістю OMA є можливість визначення пошкодження лопаті без відомого збудження а також оскільки власні

частоти залежать лише від маси та жорсткості, то цей метод не є залежним від змінних умов таких як швидкість та напрям вітру. При тестовій реалізації визначено потенціал використання ОМА для НК ВЕУ. Перспективами подальших досліджень є локалізація на класифікація типів пошкоджень.

Список літератури

1. Eldeeb, A. E., El-Arabi, M. E., Hussein, B. A. (2020). Effect of cracks in wind turbine blades on natural frequencies during operation. Journal of engineering and applied science, 67,1995–2012.https://www.researchgate.net/publication/348301648_EFFECT_OF_CRACKS_IN_WIND_TURBINE_BLADES_ON_NATURAL_FREQUENCIES_DURING_OPERATION

2. M. A. Fremmelev, Purim Lapdli, E. Orlowitz, L. O. Bernhammer, M. McGugan, K. Branner. Structural health monitoring of 52-meter wind turbine blade: Detection of damage propagation during fatigue testing. // Cambridge University Press. Data-Centric Engineering – 2022 – Vol. 3 – DOI: https://doi.org/10.1017/dce.2022.20

3. R.Henderson, Fae Azhari, A. Sinclair. Natural Frequency Transmissibility for Detection of Cracks in Horizontal Axis Wind Turbine Blades // Sensors – 2024 – 4456 – DOI: https://doi.org/10.3390/s24144456

4. Ambroise Cadoret. Operational Modal Analysis (OMA) for wind turbines health monitoring. Mechanics [physics]. Université de Rennes, 2023. English. NNT : 2023URENS046

Бадзь В.М.(1), Мельник Є.А.(2)
(1) аспірантка 3 року навчання, каф. АСУ, ІКНІ, НУЛП,
(2) студент 4 року навчання, каф. АСУ, ІКНІ, НУЛП
Науковий керівник д.т.н., професор Теслюк В.М., каф. АСУ, ІКНІ, НУЛП

ПІДХІД ДО ВИЗНАЧЕННЯ АВТОРСТВА АНГЛОМОВНИХ ТЕКСТІВ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

Вступ. У сучасну цифрову епоху, коли обсяги текстового контенту в мережі стрімко зростають, зростає й потреба в ефективних методах визначення авторства. Встановлення автора тексту має ключове значення не лише для захисту прав інтелектуальної власності, але й для забезпечення кібербезпеки, дотримання академічної доброчесності та управління цифровими публікаціями. Традиційні методи атрибуції зазвичай базуються на ручному виокремленні характеристик тексту, що може бути як трудомістким, так і малоефективним при роботі з великими та неоднорідними наборами даних. З розвитком технологій штучного інтелекту, особливо глибинного навчання, зросла увага до згорткових нейронних мереж (CNNs, Convolutional Neural Networks), які відзначаються високою ефективністю у задачах розпізнавання шаблонів, зокрема при класифікації зображень. Завдяки здатності автоматично виявляти складні структури у даних, CNNs відкривають нові можливості для автоматизованого визначення авторства, забезпечуючи вищу точність і кращу масштабованість порівняно з класичними підходами. Це дослідження присвячене створенню та аналізу підходу на основі CNNs для авторської атрибуції англомовних текстів. Особлива увага приділяється тому, як CNN-моделі здатні захоплювати стилістичні особливості авторів і перевершують традиційні методи при роботі з великими корпусами тексту та багатокласовою класифікацією. Окрім того, в роботі розглядаються можливості практичного використання цієї технології у різних галузях.

Постановка задачі. У світі сучасних технологій, де інформація зберігається онлайн, а різні засоби оцифрування паперових ресурсів набувають популярності, проблема втрати даних про автора створеного тексту стає все актуальнішою. Тому сучасні підходи до використання штучних нейронних мереж, які демонструють ефективність у різних сферах застосування, для розпізнавання авторських стилів і визначення авторства тексту все більше знаходять своє застосування і розвиток у цій предметній області. Використання згорткових нейронних мереж для задач розпізнавання та класифікації зображень показує високу ефективність порівняно з

іншими підходами [1, 2, 3, 4]. Таким чином, виникла ідея використати згорткові нейронні мережі для вирішення проблеми визначення авторства текстів.

Об'єктом дослідження є процес визначення авторства англомовних текстів із використанням CNNs. Предмет дослідження — моделі автоматизації процесу атрибуції авторства за допомогою згорткових нейронних мереж. Актуальність досліджень атрибуції авторства на основі CNNs виходить далеко за межі лінгвістики та літератури і охоплює такі сфери, як кібербезпека, правоохоронна діяльність, академічна доброчесність та цифрове видавництво. Зі зростанням значущості цифрового контенту потреба в надійних методах атрибуції авторства стає все нагальнішою, і CNNs знаходяться на вершині цього розвитку. Наукова новизна підходу на основі CNNs полягає у використанні потужної архітектури глибокого навчання для автоматичного виявлення складних стилістичних шаблонів, адаптивності до великих наборів даних і складних текстів, а також у потенціалі розширити атрибуцію авторства на інші області, такі як виявлення міжмовного плагіату, кібербезпека та правоохоронна діяльність. Це представляє значний крок вперед порівняно з традиційними методами, що базуються на ручному виділенні ознак та обмежених наборах даних.

Основний матеріал. Методи та матеріали, що використовуються для визначення авторства англомовних текстів на основі згорткових нейронних мереж, включають кілька ключових етапів, таких як підготовка даних, архітектура моделі, навчання та оцінка. Модель для визначення авторства тексту на основі згорткових нейронних мереж є сучасним підходом у галузі обробки природної мови. Цей метод використовує потужні можливості CNNs для захоплення та аналізу локальних патернів у текстових даних, що може бути корисним для виявлення стилю автора. Основні етапи та компоненти такої моделі включають: підготовку даних, векторизацію тексту, архітектуру згорткової нейронної мережі, навчання та оцінку моделі, налаштування й оптимізацію моделі (спрощена діаграма компонентів моделі представлена на рисунку 1). На етапі підготовки даних необхідно зібрати корпус текстів різних авторів (це можуть бути статті, книги, блоги чи інші типи текстів); після цього тексти аналізуються та очищаються: видалення зайвих символів, видалення стоп-слів, токенізація тексту, зведення всього тексту до нижнього регістру; потім тексти діляться на навчальну та тестову вибірки для перевірки якості моделі. Другий етап: використання методів векторизації для перетворення слів у вектори; побудова послідовностей векторів — перетворення тексту в послідовності векторів, що представляють кожне слово. Третій етап: розробка архітектури CNN. Вхідний шар приймає послідовність векторів. Згорткові шари використовують фільтри для виявлення локальних патернів у тексті; декілька згорткових шарів можуть застосовуватися для глибшого аналізу. Щільні шари відповідають за об'єднання ознак, виділених згортковими шарами, та прийняття фінального рішення. Вихідний шар: зазвичай softmax для багатокласової класифікації — визначення автора з набору можливих авторів. Для реалізації моделі CNN використовується Python через його широкі бібліотеки для машинного навчання та обробки природної мови. TensorFlow надає високорівневий API для побудови та навчання моделей CNN. NLTK є корисним для попередньої обробки і токенізації тексту. Scikit-learn використовується для оцінки метрик моделі, таких як точність, повнота, F1-міра та матриця змішування.

Результати дослідження щодо використання підходу на основі CNNs для визначення авторства англомовних текстів показали як точність, так і практичну цінність. CNNs ефективно захоплюють стилістичні нюанси за допомогою згорткових фільтрів, які чудово розпізнають n-грами (послідовності слів), унікальні для певних авторів. Це дає змогу покращити атрибуцію в довгих текстах, таких як статті, есе та романи. Створена модель на основі CNN добре працює в умовах багатокласової класифікації, де є багато можливих авторів, досягаючи вищої точності порівняно з традиційними методами. На відміну від традиційних методів, що потребують ручного виділення ознак, CNNs автоматично навчаються та виділяють релевантні ознаки з необробленого тексту, що забезпечує більш узагальнені та масштабовані моделі. Підхід на основі CNNs для визначення авторства вимагає поєднання добре підготовлених наборів даних, архітектури глибокого навчання та ретельних методів оцінки. Використовуючи здатність CNNs виявляти шаблони, цей метод дозволяє досягати точнішої атрибуції авторства, особливо у великих і складних корпусах.

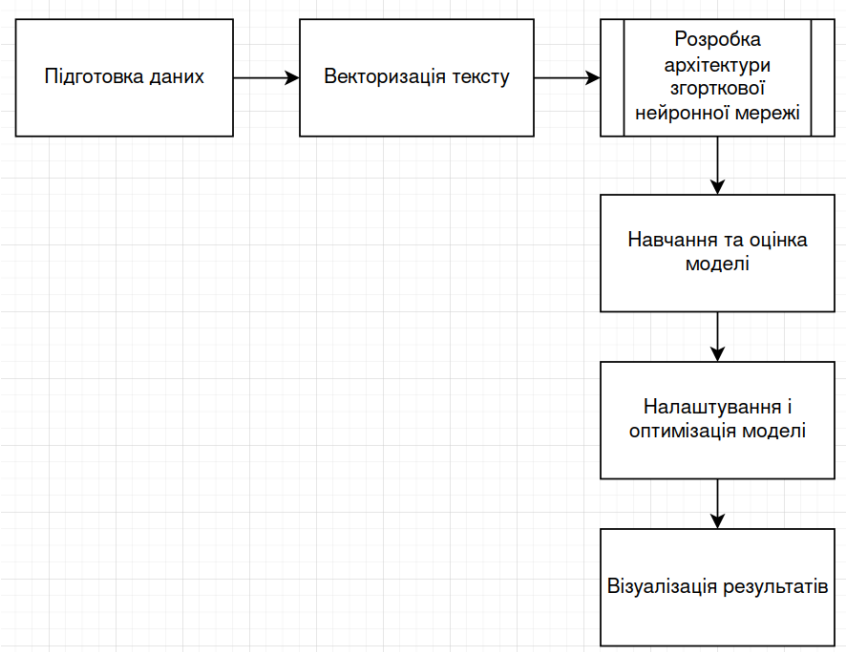


Рисунок 1 – Основні компоненти системи визначення авторства з використанням згорткових нейронних мереж

Модель на основі CNN демонструє значне поліпшення точності порівняно з традиційними методами визначення авторства. Дослідження показало, що підхід на основі CNN, завдяки своїй здатності виявляти локальні патерни в текстових даних, є особливо ефективним в умовах багатокласової класифікації, коли розглядається багато можливих авторів. Модель CNN досягла показника точності близько 90% (один з прикладів результатів моделювання представлений на рисунку 2) при ідентифікації авторів з набору можливих кандидатів, що перевершує точність, яку зазвичай досягають класичні методи, такі як наївний баєсівський класифікатор або метод опорних векторів (SVM), де вона часто становить 70–85% залежно від розміру та складності набору даних. Згорткові шари в CNN відмінно виявляють n-грами (послідовності з 3-5 слів), що є унікальними для стилю написання автора. Ця здатність допомогла моделі ефективно розрізняти стилі написання на основі тонких патернів у вживанні слів, структурі речень і синтаксисі. Модель змогла автоматично навчитися цим патернам без ручного виділення ознак, що підвищило її адаптивність та ефективність для різних жанрів тексту.

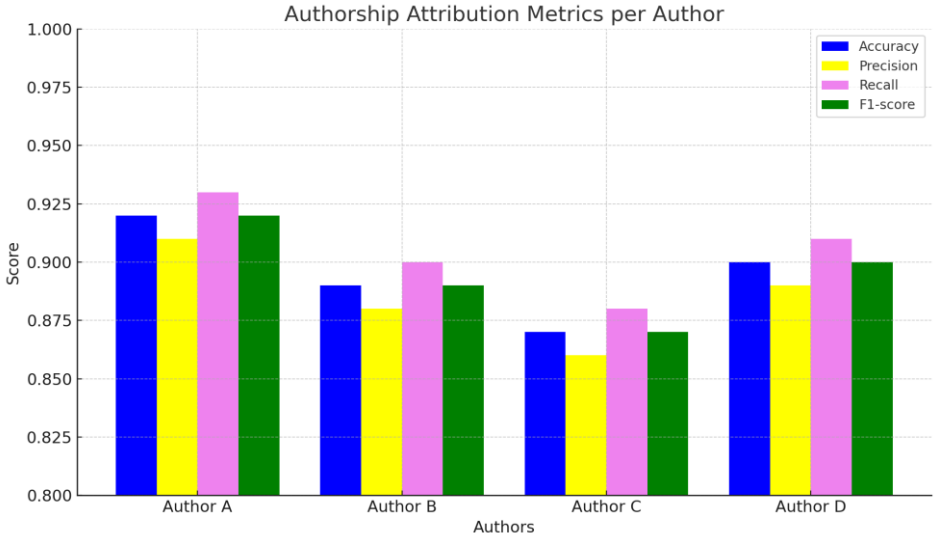


Рисунок 2 – Результати моделювання авторської атрибуції англомовних текстів представлені графічно на основі метрик для груп авторів

CNNs виявилися особливо ефективними в аналізі довгих текстів, таких як есе, статті та романи. Це відрізняється від традиційних підходів, які, як правило, втрачають точність при збільшенні довжини тексту через обмеження в процесі виділення ознак. У дослідженні модель тестувалася на текстах обсягом від 500 до 5000 слів і зберігала свою ефективність для різних розмірів даних. Створену модель на основі CNN можна використовувати для виявлення плагіату шляхом перевірки відповідності стилю написання тексту заявленому автору. Це має особливу цінність для академічних і освітніх установ, де інструменти для виявлення плагіату можуть бути покращені завдяки здатності CNN розрізняти тонкі стилістичні особливості. У сфері кібербезпеки та правоохоронної діяльності CNNs можуть використовуватися для атрибуції авторства анонімних текстів, таких як повідомлення, пости або листи з погрозами в мережі Інтернет. Це допомагає у кримінальних розслідуваннях і цифровій криміналістиці, пов'язуючи анонімні повідомлення з відомими особами на основі їхнього стилю написання. CNNs також мають практичну цінність у літературних дослідженнях для вирішення суперечок щодо авторства історичних текстів. Вони можуть допомогти в ідентифікації справжнього походження онлайн-новинних статей, розрізняючи справжні новини і фейкові, аналізуючи стиль написання автора.

Висновки. Розроблено і реалізовано підхід для визначення авторства англомовних текстів із використанням згорткових нейронних мереж, наведено основні елементи системи та їх характерні особливості. Проведено аналіз запропонованого підходу та визначено основні переваги й недоліки. Модель для визначення авторства текстів на основі згорткових нейронних мереж є потужним інструментом у сучасній сфері обробки природної мови, здатним виявляти тонкі стилістичні особливості авторів у текстах. За умови правильної підготовки даних та налаштування моделі CNNs можуть значно перевершувати традиційні методи аналізу тексту у задачах атрибуції авторства. CNNs пропонують ефективний інструмент для атрибуції авторства завдяки здатності виділяти й розпізнавати складні шаблони в текстах. Однак вони потребують добре структурованого та досить великого набору даних для навчання.

Список літератури

1. Leiyu Chen, Shaobo Li, Qiang Bai, Yang Jing, Sanlong Jiang, Yanming Miao (2022). Review of Image Classification Algorithms Based on Convolutional Neural Networks. https://www.researchgate.net/publication/357974255_Image_Classification_using_Convolutional_Neural_Networks

2. Mahbub Hussain, Jordan J. Bird & Diego R. Faria (2018). A Study on CNN Transfer Learning for Image Classification. Advances in Computational Intelligence Systems, pp 191–202. https://link.springer.com/chapter/10.1007/978-3-319-97982-3_16

3. Convolutional Neural Networks for Authorship Attribution of Short Texts (2017). Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp 669–674. https://doi.org/10.18653/v1/E17-2106

4. Purwono Purwono, Alfian Ma'arif, Wahyu Rahmaniar, Haris Imam Karim Fathurrahman, Aufaclav Zatu Kusuma Frisky, Qazi Mazhar ul Haq (2023). Understanding of Convolutional Neural Network (CNN): A Review. https://doi.org/10.31763/ijrcs.v2i4.888

Хічій А.М.
студентка 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Савка Н.Я., кафедра КІ ЗУНУ

МОДУЛЬНІСТЬ У СТИЛЯХ: ОСОБЛИВОСТІ ПОБУДОВИ МАСШТАБІВ ДИЗАЙН-СИСТЕМИ

Вступ. Із стрімким розвитком інформаційних технологій й зростанням попиту на вебзастосунки у сучасній веброзробці зростає потреба у побудові масштабованих та підтримуваних інтерфейсів. Адже із великими даними вебзастосунки стають дедалі більшими, складнішими та орієнтованими на довготривале використання. З розвитком технологій та розширенням функціональності продуктів, інтерфейс повинен адаптуватися до нових вимог без втрати цілісності та зручності. Це потребує не лише технічної гнучкості, а й візуальної та функціональної узгодженості, що неможливо досягти без чіткої структури та стандартизації. Зі збільшенням складності проєктів виникає необхідність у використанні дизайн-систем, які забезпечують узгодженість, повторне використання компонентів та спрощують командну роботу. Адже дизайн-система допомагає структурувати підхід до розробки, підвищує ефективність та покращує користувацький досвід. Без єдиних стандартів інтерфейс швидко втрачає цілісність: елементи дублюються, стилі плутаються, а підтримка проєкту ускладнюється.

Існуючі підходи до розробки дизайн-систем, зазвичай, пропонують ускладнені рутинні механізми, що робить їх непридатними для масштабованих систем. Tailwind CSS, як utility-first фреймворк, пропонує ефективний інструмент розробки таких систем завдяки своїй модульності, конфігурованості та можливості швидкого прототипування [1], що робить його особливо корисним для сучасних інтерфейсних рішень.

Постановка задачі. Об'єкт дослідження – дизайн-системи у веброзробці. Предмет дослідження – побудова модульних та масштабованих дизайн-систем на основі Tailwind CSS. Мета дослідження – проаналізувати можливості Tailwind CSS для розробки повторно використовуваних компонентів та оцінити ефективність інструменту для побудові масштабованих дизайн-систем.

Основний матеріал. У процесі дослідження проаналізовано особливості використання Tailwind CSS для побудови масштабованих дизайн-систем, зокрема у контексті розробки вебзастосунку для організації подорожей. Такий застосунок передбачає можливість зберігати власні маршрути, бронювання, квитки та інші супровідні дані, що вимагає не лише високого рівня зручності користувацького інтерфейсу, але й його гнучкої адаптації до різних сценаріїв використання.

Tailwind CSS, як utility-first фреймворк, орієнтований на розробку модульних компонентів без необхідності писати окремі стилі у CSS-файлах. Замість цього розробники використовують готові класи безпосередньо в HTML, що дозволяє швидко компонувати інтерфейс [1]. Це суттєво спрощує процес розробки, особливо коли йдеться про реалізацію численних елементів інтерфейсу, зокрема, картки маршрутів, списки бронювань, форми пошуку рейсів, квитків або готелів. Завдяки своїй структурі, Tailwind сприяє швидкому прототипуванню та знижує кількість повторюваного коду. Крім того, він забезпечує гнучкість при масштабуванні: кожен компонент ізолюється у межах свого шаблону, легко модифікується без впливу на інші частини системи, що особливо важливо в багатофункціональних та динамічних проєктах.

Ключова перевага Tailwind CSS – централізована система токенів у файлі tailwind.config.js, де в одному місці задаються кольори, шрифти, відступи, брейкпоінти та інші стилі [2]. Це суттєво спрощує підтримку візуальної узгодженості та дозволяє легко змінювати глобальні стилі без редагування кожного компонента окремо. У класичному підході, наприклад, із SCSS та методологією BEM (Block Element Modifier), централізація досягається через змінні SCSS, але вона менш гнучка – потребує чіткої структури класів, контролю вкладеності та імен. BEM вважають стандартом побудови масштабованого CSS, однак вимагає більше ручної роботи й

має вищий поріг входження. Tailwind CSS у цьому контексті пропонує гнучкіший та прозоріший підхід завдяки системним класам і єдиній конфігурації.

Окрім цього, можливість використовувати директиву @apply у Tailwind CSS дозволяє винести повторювані стилі в окремі кастомні класи, такі як .btn-primary або .card-tour, що містять набір utility-класів [3]. Це дозволяє не лише централізовано змінювати стиль компонентів, але й суттєво знижує дублювання коду. У проєкті, де користувач взаємодіє з великою кількістю подібних за структурою елементів (наприклад, перелік бронювань або картки з інформацією про подорож), така перевага відіграє важливу роль у підтримці чистоти коду та його повторного використання. Аналогічний підхід у SCSS реалізується за допомогою міксинів або наслідування, однак їхнє впровадження є складнішим та менш інтуїтивним.

Порівняння підходів здійснено на прикладі реалізації картки туристичного маршруту. При використанні Tailwind усі стилі задавалися безпосередньо у шаблоні компонента Vue, що забезпечило повну ізольованість компонента та мінімізувало залежність від глобальних стилів. Завдяки цьому компонент є компактним, легко підтримується та швидко модифікується. Натомість, при реалізації аналогічного елемента з використанням підходу BEM довелося розробляти окрему структуру класів – .card, .card__title, .card__description, і вручну прописувати відповідні стилі в SCSS-файлах. Це призвело до збільшення обсягу CSS-коду, ускладнило організацію стилів та потребувало додаткової уваги до іменування і структури. Tailwind у цьому контексті продемонстрував вищу ефективність у розробці дрібних, багаторазових UI-компонентів, де важлива швидкість і гнучкість.

Ще одна важлива перевага Tailwind – підтримка адаптивності через вбудовані медіа-запити в utility-класах. Це дозволяє швидко розробляти інтерфейси, які добре працюють на різних пристроях – від смартфонів до великих екранів. Для вебзастосунків, якими часто користуються в дорозі, це особливо цінно. На відміну від цього, у SCSS медіа-запити прописуються вручну, що ускладнює структуру коду та уповільнює розробку [4].

Також варто зазначити й деякі недоліки Tailwind. На етапі початкової розробки шаблони HTML можуть виглядати перевантаженими через велику кількість utility-класів, що ускладнює читабельність коду при побіжному перегляді. Проте ці труднощі, зазвичай, компенсуються у процесі розвитку проєкту завдяки високій структурованості, передбачуваності поведінки компонентів та загальній масштабованості підходу [5].

У підсумку, Tailwind CSS виступає сучасним і ефективним інструментом для побудови гнучких дизайн-систем у багатофункціональних вебзастосунках. У випадку проєкту для організації подорожей, він дає змогу швидко реалізовувати складні інтерфейси, адаптувати їх під різні пристрої, підтримувати візуальну узгодженість і суттєво пришвидшити розробку без втрати якості.

Висновки. Tailwind CSS є ефективним інструментом для створення модульних і масштабованих дизайн-систем. Підхід utility-first сприяє повторному використанню компонентів, узгодженості дизайну та спрощенню підтримки коду. Використання @apply, кастомних класів і конфігурації теми забезпечує гнучкість і швидкість розробки. Tailwind особливо доречний для динамічних застосунків, як-от сервіси для організації подорожей. Попри початкову складність, фреймворк значно полегшує масштабування й розвиток інтерфейсів у довгостроковій перспективі.

Список літератури

1. Tailwind CSS Documentation: Reusing Styles. Tailwind CSS: вебсайт. URL: https://v3.tailwindcss.com/docs/reusing-styles (дата звернення: 12.05.2025).

2. Tailwind CSS Documentation: Adding Custom Styles. Tailwind CSS: вебсайт. URL: https://tailwindcss.com/docs/adding-custom-styles (дата звернення: 12.05.2025).

3. Pieces.app Blog: Tailwind @Apply: Replacing Complex Classes with Tailwind CSS. Pieces: вебсайт. URL: https://pieces.app/blog/tailwind-apply-css-replacing-complex-classes (дата звернення: 12.05.2025).

4. Використання Tailwind CSS для швидкого розробки стильового дизайну: вебсайт. URL: https://it-rating.ua/vikoristannya-tailwind-css-dlya-shvidkogo-rozrobki-stilovogo-dizaynu (дата звернення: 12.05.2025).

5. Andrews J. "Utility-First Frameworks in Practice: A Comparative Analysis of Tailwind CSS and Traditional CSS Methodologies // Journal of Front-End Engineering." 2022. p. 68–79.

Когуч О.Г.(1), Загородний І.І.(2)
аспіранти 2 року навчання каф. СШІ ІКНІ НУ«ЛП»
Науковий керівник д.т.н., професор Матвійчук Я.М., кафедра СШІ ІКНІ НУ«ЛП»

ШТУЧНИЙ ІНТЕЛЕКТ У СУЧАСНИХ ЗАСОБАХ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ. Упродовж останнього десятиліття, зокрема в останні роки, досягнення у сфері штучного інтелекту (ШІ) зумовили переосмислення класичних практик програмної інженерії, зокрема в аспектах автоматизації, інтерактивної генерації коду та гнучкості моделей взаємодії між розробником і інструментом [1, 3].

Впровадження інтелектуальних інструментів у середовища розробки сприяє автоматизації широкого спектра завдань [2]: від проєктування, генерації, автодоповнення та рефакторингу (структурного вдосконалення коду без зміни його функцій) коду до його тестування, документації, інтеграції API та супроводу. Такі інструменти не лише підвищують продуктивність і якість коду, а й знижують когнітивне навантаження на розробника, скорочують витрати часу і ресурсів, а також суттєво зменшують періоду від початку розробки продукту до його виходу на ринок (time-to-market).

У цьому контексті особливої актуальності набуває дослідження вимог до сучасних інтегрованих середовищ розробки (IDE) та їхньої адаптації до потреб використання ШІ-компонентів. Зокрема, виникає потреба в комплексному аналізі моделей інтеграції ШІ у життєвий цикл створення програмного забезпечення (SDLC) як на рівні щоденних задач окремого інженера, так і в рамках командної розробки за усталеними SDLC-практиками.

Постановка задачі. Об’єктом дослідження є інструменти, що застосовуються на різних етапах життєвого циклу розробки програмного забезпечення від написання коду до його тестування, документування та розгортання.

Предметом дослідження є інтелектуальні компоненти, інтегровані в ці інструменти, зокрема: ШІ-асистенти, автономні ШІ-агенти, великі мовні моделі (LLM), які реалізують функції генерації, аналізу, оптимізації та супроводу коду. Ці компоненти інтегруються в сучасні IDE, а також у сервісно-орієнтовані платформи, побудовані на базі ШІ-технологій.

Метою роботи є аналіз сучасних підходів до інтеграції ШІ-компонентів у інструменти розробки ПЗ, виявлення їх ключових функцій, а також формулювання узагальнених вимог до таких систем з орієнтацією на ефективність, адаптивність і відповідність потребам розробника.

Основний матеріал. На поточному етапі розвитку програмної інженерії інтегровані середовища розробки (IDE) дедалі частіше доповнюються вбудованими або модульними ШІ-компонентами, що орієнтовані на підтримку розробника під час виконання як рутинних, так і складних інженерних задач.

Серед таких середовищ доцільно розрізняти два типи: класичні IDE, які отримали ШІ-функціональність шляхом розширень або інтеграції з хмарними LLM-сервісами (напр., Visual Studio, VS Code, IntelliJ IDEA, PyCharm, NetBeans, WebStorm, Xcode, Eclipse) [2]; редактори нового покоління, які з самого початку були спроєктовані як ШІ-орієнтовані системи з глибокою інтеграцією мовних моделей на всіх етапах роботи з кодом (напр., Cursor, WindSurf) [4, JetBrains Blog, InfoQ].

Механізми інтеграції можуть реалізовуватись у двох формах: ШІ-асистенти (LLM-інструменти, що діють безпосередньо у вікні редактора та реагують на запити користувача (написання функцій, пояснення, автодоповнення)) та агентні системи (автономні підсистеми, які виконують багатокрокові задачі (генерація модулів, тестів, реструктуризація коду тощо) на основі поставленої мети).

Серед найбільш розповсюджених ШІ-асистентів можна назвати GitHub Copilot, JetBrains AI Assistant, Amazon Q Developer, Tabnine, AskCodi, Codiga, Replit Ghostwriter, Intellicode, Figstack, OpenAI Codex, Sourcegraph Cody, DeepCode AI, CodeT5, CodeGeeX та ін. [2, 3]. Ці інструменти розрізняються за фокусом функціональності: одні спеціалізуються на генерації коду, інші на рефакторингу, поясненні логіки, автоматизованому тестуванні або документації.

За рахунок контекстної обробки коду, підказок у реальному часі та знань із відкритих репозиторіїв (сховищ коду), ШІ-асистенти суттєво підвищують продуктивність, зменшують кількість помилок та підтримують стандартизацію стилю програмування у команді.

На відміну від асистентів, ШІ-агенти мають вищий рівень автономії та здатні виконувати багатокрокові дії на основі поставлених задач [4]. Вони не просто реагують на запити, а формують власні плани дій, координують кроки та адаптують свою поведінку відповідно до контексту. У процесі взаємодії з кодом такі агенти можуть проводити аудит існуючих фрагментів, модифікувати структуру програмного модуля, автоматично інтегрувати зовнішні API, або навіть взаємодіяти з іншими агентами в системі. Серед практичних прикладів реалізації агентних рішень можна виділити: Cursor і Windsurf (Codeium) - із глибокою інтеграцією планувальників на основі LLM; JetBrains Junie - як експериментальний агент у середовищі IntelliJ; Google Jules і Replit Agent v2 - орієнтовані на автоматизацію типових завдань у хмарному середовищі; Theia Coder - вбудований агент для платформи Eclipse Theia [JetBrains Blog, InfoQ, Replit Blog, Theia Docs].

Паралельно активно розвиваються фреймворки (англ. *frameworks*, каркаси, інструментальні платформи) для побудови адаптованих агентних систем (LangChain, LangGraph, AutoGen (Microsoft), Semantic Kernel, OpenAI Agents SDK (Swarm), SuperAGI, CrewAI) [4, GitHub Docs], які дозволяють реалізовувати розширені сценарії автоматизації розробки від генерації багатомодульного коду до координації командної роботи в межах IDE або CI/CD-процесу.

Функціональність ШІ-асистентів та агентів безпосередньо визначається мовними моделями, на базі яких вони працюють. Саме великі мовні моделі (LLM) забезпечують здатність таких систем розуміти програмний код, формувати нові фрагменти, пояснювати логіку, оптимізувати структуру, виявляти помилки та підтримувати технічну комунікацію з користувачем. Станом на травень 2025 року серед найбільш ефективних LLM для задач програмування вирізняються:

1. OpenAI GPT моделі: GPT-4.1 (реліз 14.04.2025) - характеризується високою точністю, глибоким контекстним розумінням, здатністю до обробки складних ланцюгів інструкцій; оптимізована версія GPT-4.1-mini, яка рекомендована для інтеграції в IDE. OpenAI o3/o3-mini-high/o4-mini, що застосовуються у сучасних агентних фреймворках завдяки покращеному плануванню, підтримці короткострокової пам’яті та адаптивності в багатокрокових сценаріях.

2. Anthropic Claude моделі: Claude 3.x: Claude 3.7 Sonnet і 3.5 Sonnet ефективні в рефакторингу та стислому поясненні коду; 3.5 Haiku - малоресурсна модель із високою якістю відповідей.

3. Google Gemini 2.5 моделі: Pro-Exp-03-25 - мультимодальна модель із підтримкою тексту, коду та зображень; Flash-Preview-04-17 - оптимізована для швидкої генерації у хмарних середовищах.

Окремої уваги заслуговують нові моделі від DeepSeek, зокрема DeepSeek-V3-0324 та DeepSeek-R1, які спеціалізуються на високоточному аналізі коду та демонструють конкурентну якість у задачах автодоповнення, автотестування та узагальнення шаблонних патернів [5]. У сегменті експериментальних LLM-рішень відзначається early-grok-3 - модель, що попри обмежену доступність, активно тестується в межах відкритих проєктів і дослідницьких платформ (її особливістю є спроба поєднати генерацію коду з мультимодальними можливостями).

Загалом, різноманіття сучасних LLM-моделей створює широкий простір для адаптації розробницьких інструментів до потреб конкретних проєктів і команд. Це дозволяє обирати оптимальний баланс між якістю генерації, швидкодією, вартістю викликів і доменною спеціалізацією моделей.

Варто підкреслити, що екосистема великих мовних моделей розвивається надзвичайно динамічно. Оновлення параметрів, запуск експериментальних версій, оптимізація затримок та розширення контексту відбуваються щотижня. Це безпосередньо впливає на якість, стабільність і відповідність результатів, які отримують інженери. Тому систематичне відстеження стану моделей є критично важливим при ухваленні рішень щодо їх інтеграції у життєвий цикл програмного забезпечення. Для цього існують спеціалізовані платформи моніторингу, які

надають актуальну аналітику, рейтинги та порівняльні таблиці продуктивності LLM у задачах програмування: Papers with code - систематизовані результати досліджень на наборі даних HumanEval; aider.chat - огляд продуктивності у середовищах генерації коду; openrouter.ai - рейтинг моделей за швидкодією, точністю та доступністю API; vellum.ai - оцінка моделей у прикладних сценаріях; convex.dev – порівняння по переліку показників; EvalPlus – рейтинг за метриками точності в задачах кодування. Інформація про продуктивність та специфіку застосування моделей регулярно оновлюється на зазначених платформах.

В умовах стрімкого розвитку LLM-технологій, інтегровані середовища розробки (IDE) повинні адаптуватися до нових можливостей мовних моделей і водночас зберігати вимоги до стабільності, ефективності та безпеки. Це потребує чіткого визначення функціональних критеріїв для інтелектуальних середовищ. Така формалізація дозволяє: зіставляти різні IDE за рівнем підтримки ШІ, виявляти слабкі місця у впровадженні агентів або асистентів, і краще орієнтуватися при виборі платформи для команд розробників [2, 3, 4]. Нижче наведено таблицю 1, в якій систематизовано функціональні характеристики, які були опрацьовані авторами в рамках дослідження, рекомендовані для реалізації в сучасних IDE з урахуванням використання LLM та агентних компонентів.

Таблиця 1. Основні функціональні характеристики інтегрованого середовища розробки при використанні інтелектуальних компонент на базі систем штучного інтелекту

№	Категорія	Функціональні можливості
1	Генерація коду	Створення коду «з нуля», генерація коду з природної мови, рефакторинг наявного коду, генерація фрагментів коду, приклади коду для конкретних API, перетворення коментарів у код, генерація файлів налаштувань.
2	Точність генерації коду	Мінімальна кількість програмних і синтаксичних помилок, коректне використання бібліотек, правильна алгоритмічна логіка, урахування мовних особливостей, генерація з урахуванням архітектури проєкту, дотримання стандартів кодування, автоматичне вирішення залежностей.
3	Контекстна обізнаність	Розуміння контексту всього проєкту, відстеження області видимості змінних/функцій, розв’язання міжфайлових посилань, використання пам’яті мовної моделі, збереження стану сесії (за наявності), автоматичне отримання відповідної документації, обробка імпортованих бібліотек/модулів, підтримка логічної цілісності коду.
4	Розуміння природної мови	Інтерпретація та уточнення складних інструкцій, обробка багатокрокових запитів, переклад галузевої термінології, аналіз коментарів, розрізнення типів завдань (виправлення, створення, тестування, тощо), розуміння форматування та стилю найменування, підтримка багатомовного вводу.
5	Автодоповнення та підказки в коді	Підказки з урахуванням контексту, прогнозування кількох токенів наперед, підказки з урахуванням типів, генерація без затримки, завершення з урахуванням синтаксису, оцінювання та впорядкування пропозицій.
6	Перевірка коду	Виявлення програмних помилок і вразливостей, пропозиції з оптимізації, позначення шаблонів хибних рішень (антипатернів), надання обґрунтувань, підтримка перевірки безпеки, поведінка подібна до статичних аналізаторів коду, пропозиції щодо стилю, перехресне посилання на документацію.
7	Підтримка кількох файлів	Читання та запис у кількох файлах, відстеження імпортів/експортів, поширення конфігурацій, робота з великими кодовими базами, розумні пропозиції щодо файлів, врахування змін, перейменування/рефакторинг на рівні проєкту, побудова графу залежностей.
8	Розуміння команд терміналу	Виконання команд CLI, пропозиції правильного синтаксису, перетворення тексту на shell-команди, навігація файловою структурою, інтеграція з Git, пояснення виводу терміналу, автоматизація ланцюжків команд.
9	Можливості ШІ-асистента	Взаємодія в реальному часі, активне стеження за контекстом і структурою проєкту, автоматичне уточнення завдань, ініціація дій на основі інструкцій, рольова поведінка, адаптація до стилю користувача або проєкту, запам’ятовування попередніх інструкцій у межах сесії.

№	Категорія	Функціональні можливості
10	Можливості ШІ-агента	Планування багатокрокових завдань, взаємодія з пам’яттю, моделювання середовища, відстеження прогресу, логічне ланцюгове мислення, декомпозиція задач, відновлення після помилок і самодіагностика.
11	Безпечна розробка	Уникнення небезпечних практик, виявлення закодованих секретів, виявлення вразливостей до ін’єкцій, безпечні налаштування за замовчуванням, інтеграція з безпековими інструментами, попередження про небезпечні бібліотеки, усвідомлення дозволів і прав доступу.
12	Налаштування стилю розробки	Підлаштування під особистий стиль, правила для конкретного проєкту, користувацькі шаблони запитів, конфігурація плагінів/модулів, збереження уподобань користувача, поведінка за ролями/завданнями, бібліотека кодових фрагментів, визначення власних команд.
13	Перетворення візуального макету в код	Figma в HTML/CSS або JS/TS, виявлення макетів, витягнення стилів, генерація адаптивного дизайну, компонентно-орієнтований підхід, підтримка CSS/JS фреймворків, підказки щодо доступності, виявлення повторного використання стилів/класів.
14	Автоматичне створення коментарів	Створення коментарів до фіксацій змін у системі контролю версій (англ. *commits*): опис змін у файлах, підсумок логіки коду, відповідність шаблонам повідомлень, посилання запиту на злиття, підтримка стандартизованих фіксацій змін, зведення змін у кількох файлах, пропозиції до історії змін.
15	Інтеграція з інструментами та API для розробки	Підтримка GitHub, GitLab Actions, контроль CI/CD процесів, інтеграція з статичними аналізаторами коду та тестами, виклик зовнішніх API, запити до баз даних, автоматизація файлового вводу/виводу, використання зовнішніх CLI-інструментів.
16	Підтримка запитів (Prompt Engineering)	Побудова складних ланцюгів запитів, конфігурація системного запиту, інструменти для тестування запитів, історія, автоматичне створення та метрики оцінки.
17	Співпраця та парне програмування	Сесії з кількома користувачами, анотовані підказки, спільні перегляди коду, потокове обговорення пропозицій, рольові режими агента, інтеграція з Git для перевірки коду, підтримка обговорень в коді та чаті, збереження зворотного зв’язку з користувачем.
18	Пояснення	Анотації до коду, поетапна інтерпретація алгоритмічного змісту, обґрунтування змін, аналіз причин помилок, коментарі щодо продуктивності, візуальні пояснення (на прикладі), розбір помилок.
19	Допомога з тестуванням	Генерація сценаріїв тестування, виявлення крайових випадків, підказки щодо покриття тестами, підтримка модульного та інтеграційного тестування, інтеграція з тестовими фреймворками, пропозиції тестових об’єктів/заглушок, взаємодія у стилі TDD.
20	Проєктування взаємодії	Підтримка мультимодального вводу, зрозумілий та послідовний зворотній зв’язок, функція «скасувати/повторити» зміни, контекстні підказки й допоміжний текст, інтерфейс для історії команд, гарячі клавіші та командна панель, індикатори прогресу.

Наведені вище аспекти формулюють перелік ключових компонентів та вимог, яким мають відповідати сучасні інструменти для розробки програмного забезпечення з вбудованими інтелектуальними технологіями [1, 3, 4]. Саме наявність комплексної, багатовимірної підтримки (від генерації коду до його перевірки, адаптації, тестування та пояснення) визначає прикладну цінність ШІ-систем у процесі розробки. Такий підхід забезпечує реальне підвищення ефективності на всіх етапах життєвого циклу програмного продукту, скорочуючи час, знижуючи ймовірність помилок і покращуючи якість інженерних рішень.

У межах дослідження було проаналізовано низку сучасних інструментів програмної інженерії з інтегрованими LLM-компонентами. На основі цього аналізу було виокремлено типові сценарії використання, які корелюють із запропонованими функціональними характеристиками. Зокрема, часткова або повна реалізація окремих можливостей спостерігається у таких рішеннях, як GitHub Copilot, Codeium, Windsurf, а також у середовищах JetBrains, зокрема WebStorm та IntelliJ IDEA. Ці приклади підтверджують практичну цінність

представленого підходу та демонструють ефективність ШІ-асистентів у реальному контексті розробки. Нижче на рисунку 1 наведено приклад реалізації відповідних функціональностей у середовищі JetBrains.

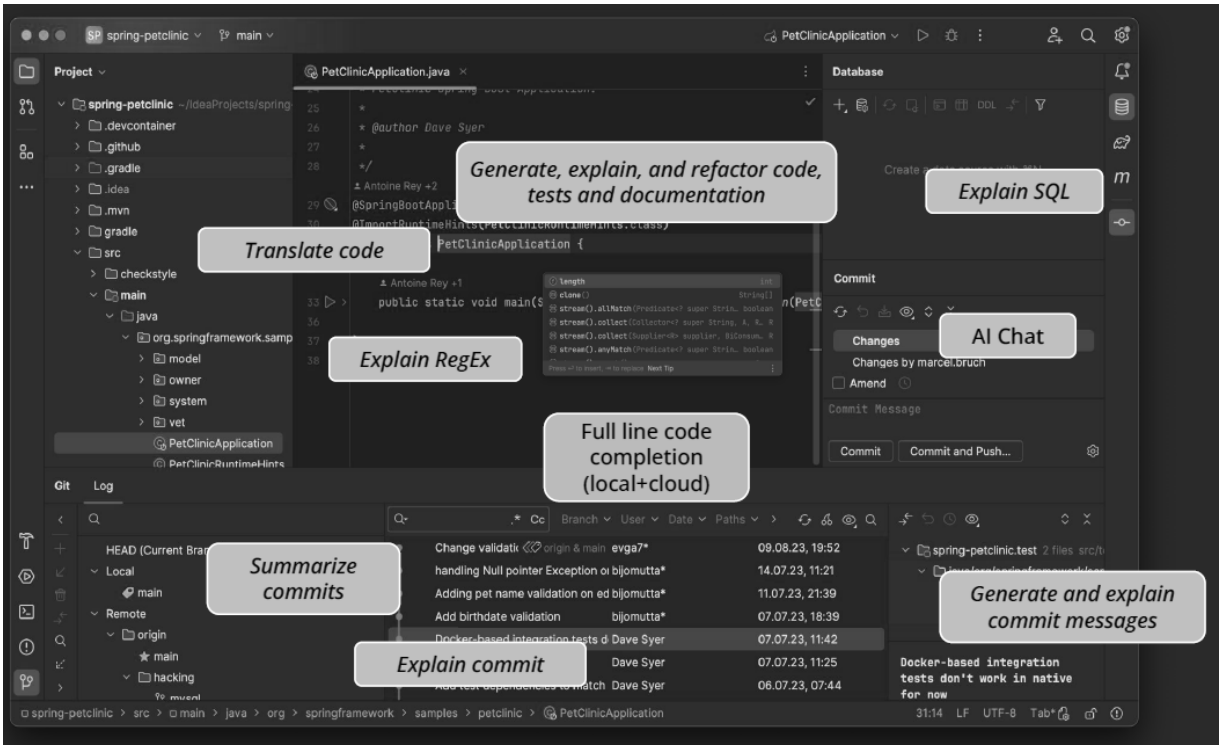


Рисунок 1. Інтеграція можливостей штучного інтелекту в середовище розробки IntelliJ IDEA. Джерело: знімок екрана середовища розробки IntelliJ IDEA з інтеграцією AI-функцій.

На зображенні представлено графічний інтерфейс середовища розробки IntelliJ IDEA з активованими функціями штучного інтелекту, які забезпечують: генерацію, пояснення та рефакторинг коду, тестів і документації (автоматизоване створення та покращення вихідного коду); переклад коду (можливість трансформації фрагментів між різними мовами програмування); пояснення регулярних виразів (спрощення розуміння складних шаблонів пошуку); повне автодоповнення коду (для підвищення продуктивності за рахунок швидкого й точного введення); пояснення SQL-запитів (допомога в аналізі та оптимізації запитів до баз даних); інтерактивний ШІ-чат (отримання відповідей на запити безпосередньо в IDE); генерацію та пояснення фіксацій змін у коді (автоматичне створення інформативних повідомлень); пояснення окремих фіксацій (аналіз змін у репозиторії); а також стислий опис історії змін (швидкий огляд внесених змін). Повний перелік функцій доступний на офіційному сайті: https://www.jetbrains.com/idea/features .

Висновки. Інтеграція інтелектуальних компонентів у процес розробки програмного забезпечення сприяє глибокій трансформації сучасної інженерної практики. Використання засобів на основі штучного інтелекту, передусім великі мовні моделі, інтелектуальні асистенти та агентні системи, забезпечують автоматизацію критичних етапів життєвого циклу програмного продукту від проєктування, аналізу, генерації, оптимізації та рефакторингу коду до його тестування, верифікації та документування. Як зазначено у [1, 2], ці інструменти не лише знижують когнітивне навантаження на розробника, а й істотно підвищують продуктивність командної роботи, сприяють дотриманню архітектурної узгодженості та скорочують час реалізації функціональних вимог. У контексті зростаючої складності програмних систем і високої динаміки змін на ринку, впровадження ШІ у середовища розробки виступає стратегічним чинником забезпечення конкурентоспроможності та інноваційності програмних рішень.

Список літератури

1. A. Sergeyuk, I. Zakharov, E. Koshchenko, M. Izadi, "Human-AI Experience in Integrated Development Environments: A Systematic Literature Review," arXiv preprint arXiv:2503.06195, 2025, pp. 1–15. https://arxiv.org/abs/2503.06195
2. S. B. Nettur, S. Karpurapu, U. Nettur, L. S. Gajja, S. Myneni, A. Dusi, "The Role of GitHub Copilot on Software Development: A Perspective on Productivity, Security, Best Practices and Future Directions," arXiv preprint arXiv:2502.13199, 2025, pp. 1–14. https://arxiv.org/abs/2502.13199
3. C. Treude, M. A. Gerosa, "How Developers Interact with AI: A Taxonomy of Human-AI Collaboration in Software Engineering," arXiv preprint arXiv:2501.08774, 2025, pp. 1–12. https://arxiv.org/abs/2501.08774
4. M. A. Ferrag, N. Tihanyi, M. Debbah, "From LLM Reasoning to Autonomous AI Agents: A Comprehensive Review," arXiv preprint arXiv:2504.19678, 2025, pp. 1–35. https://arxiv.org/abs/2504.19678
5. N. Huynh, B. Lin, "Large Language Models for Code Generation: A Comprehensive Survey of Challenges, Techniques, Evaluation, and Applications," arXiv preprint arXiv:2503.01245, 2025, pp. 1–28. https://arxiv.org/abs/2503.01245

Загородний І.І.
аспірант 2 року навчання каф. СШІ ІКНІ НУ«ЛП»
Науковий керівник д.т.н., професор Шаховська Н.Б., кафедра СШІ ІКНІ НУ«ЛП»

ВІЗУАЛІЗАЦІЯ АКУСТИЧНИХ СИГНАЛІВ ЯК АЛЬТЕРНАТИВА ТРАДИЦІЙНІЙ ЦИФРОВІЙ ОБРОБЦІ ДЛЯ МАШИННОГО НАВЧАННЯ

Вступ. Перетворення звуку на зображення є ефективним міждисциплінарним підходом, що відкриває нові можливості для обробки та аналізу акустичних сигналів який активно застосовується в науці, медицині, мистецтві та технологіях. Серед основних методів такого перетворення можна виділити спектрограми (для аналізу частотного складу сигналу), вейвлет-перетворення (для виявлення короткочасних змін у сигналі), мел-спектрограми (що враховують особливості сприйняття людиною), мел-кепстральні коефіцієнти (що використовуються в розпізнаванні мови) та інші [1,4]. Кожен із цих методів дозволяє зробити акцент на різних характеристиках сигналу - частотній структурі, динаміці змін, тембрі, гармоніках, ритміці - забезпечуючи гнучкість відповідно до конкретного завдання.

Заміна прямої обробки аудіосигналів у часовому домені на їхню візуалізацію дає змогу застосовувати широкий спектр методів глибокого навчання для обробки візуальної інформації. У сфері біомедицини перетворення звукових сигналів у зображення набуває особливої цінності. Біологічні звуки (такі як серцевий ритм, дихання, рух повітря в легенях, голосові коливання чи сигнали з внутрішньочерепної активності) мають складну, часто нелінійну структуру. Перетворення цих сигналів у візуальні репрезентації дає змогу застосовувати вже апробовані та ефективні методи аналізу зображень (наприклад, для виявлення аномалій або класифікації патологій), а також адаптувати існуючі підходи штучного інтелекту, такі як глибокі нейронні мережі чи традиційні методи машинного навчання, під задачі біо-акустичного моніторингу і аналізу [2].

Перетворення звуку на зображення має низку переваг: дозволяє застосовувати ефективні методи аналізу зображень на основі штучного інтелекту, полегшує відокремлення шумів завдяки їхній візуальній відмінності, зберігає як часову, так і частотну інформацію для точнішої класифікації. Такий підхід спрощує виявлення шаблонів, уніфікує аудіо для роботи з універсальними моделями, сприяє пояснюваності рішень через візуалізації, легко інтегрується в мультимодальний аналіз, дозволяє аугментацію даних (важливо при обмежених вибірках,

зокрема в медицині), працює в реальному часі навіть на малопотужних пристроях і полегшує інтерпретацію для непрофільних спеціалістів [1, 3].

Постановка задачі. Об'єкт дослідження - сучасні методи подання аудіоінформації у візуальній формі. Предмет дослідження - методи візуалізації акустичних сигналів (спектрограми, мел-спектрограми, вейвлет-перетворення тощо) та їх застосування у сфері біомедичного аналізу з використанням технологій штучного інтелекту. Мета дослідження - здійснити порівняльний аналіз методів перетворення звуку у зображення у контексті біоакустичного моніторингу, діагностики та автоматизованого аналізу медичних сигналів. Особливу увагу приділено перевагам цього підходу для інтеграції з моделями глибокого навчання. Гіпотеза дослідження полягає в тому, що поєднання методів перетворення звуку дозволяє підвищити точність біоакустичної діагностики порівняно з використанням кожного методу окремо.

Основний матеріал. У сучасній практиці аналізу звукових сигналів усе частіше використовуються методи візуального кодування акустичних характеристик. Такий підхід дозволяє не лише спростити сприйняття сигналу людиною, а й зробити його придатним для обробки за допомогою алгоритмів обробки візуальної інформації. У цьому дослідженні основну увагу було приділено аналізу технік, які забезпечують ефективну візуалізацію різних аспектів звуку. До числа таких технік увійшли спектрограми, мел-спектрограми, вейвлет-перетворення, а також мел-кепстральні коефіцієнти, що демонструють різні можливості з погляду глибини аналізу, адаптації до фізіологічних особливостей та практичної реалізації в медичних і технічних задачах [2].

На відміну від сирих звукових сигналів, які мають одномірну структуру та потребують складної попередньої обробки, їхні візуальні інтерпретації вже відповідають формату, з яким сумісні більшість сучасних моделей глибокого навчання. Це дає змогу використовувати існуючі архітектури, зокрема згорткові нейронні мережі, що здатні автоматично виокремлювати інформативні ознаки без ручного створення дескрипторів. Крім того, зображення забезпечують уніфікований формат вхідних даних, що спрощує застосування готових моделей і підвищує ефективність в умовах обмежених ресурсів. У біомедичних задачах це дозволяє оперативно розробляти ефективні рішення без потреби у великих спеціалізованих аудіо-датасетах. Візуальні форми, отримані зі звукових даних, створюють зручне представлення для використання моделей, оптимізованих під структуровані дані фіксованого розміру. Це дозволяє застосовувати готові рішення, здатні виявляти ключові характеристики без додаткової обробки сигналу [2]. Важливою перевагою є також можливість перенавчання: моделі, попередньо натреновані на великих наборах зображень, можна адаптувати до медичних задач навіть за умов обмеженого обсягу прикладів. Такий підхід не лише прискорює розробку, а й підвищує надійність результатів. До того ж наочне представлення даних сприяє кращій інтерпретації й перевірці рішень, що є особливо важливим у медичних системах підтримки прийняття рішень [1].

У цьому дослідженні проведено порівняльний аналіз чотирьох поширених підходів до візуального представлення звукових сигналів, які використовуються як вхідні дані для глибоких моделей машинного навчання: спектрограма, мел-спектрограма, вейвлет-перетворення (scalogram) та мел-кепстральні коефіцієнти (MFCC). Кожен із методів має свої переваги щодо збереження часово-частотної інформації, візуальної структури сигналу та сумісності з архітектурами нейронних мереж [4].

- Спектрограма - це зображення, яке показує, як змінюється частотний склад звуку з часом. Вона створюється за допомогою короткочасного перетворення Фур'є (STFT), яке розбиває сигнал на короткі вікна та аналізує кожне з них окремо. У спектрограмі горизонтальна вісь - це час, вертикальна - частота, а колір або яскравість - інтенсивність сигналу. Такий підхід дозволяє бачити зміну частот у часі й широко використовується в медицині для аналізу дихання, серцевих звуків і моніторингу пацієнтів [1, 4].
- Мел-спектрограма (MEL) - це варіант спектрограми, де частоти масштабуються відповідно до мел-шкали, яка відображає сприйняття людиною. Вона теж базується на STFT, але частотні компоненти групуються за допомогою спеціального набору фільтрів. Отримане зображення краще відповідає тому, як ми чуємо звуки. Мел-спектрограми застосовуються для

аналізу мовлення при підозрі на неврологічні розлади, розпізнавання емоційного стану та дослідження дихальних аномалій [2, 4].

- Скалограма (вейвлет-перетворення) - це графік, який показує, як змінюється структура сигналу одночасно в часі та по частоті. Метод базується на вейвлет-перетворенні, яке дозволяє виявити короткочасні події, імпульси й коливання. На скалограмі одна вісь відображає час, інша - масштаб (який відповідає частоті), а значення - енергію сигналу. У медицині скалограми використовують для аналізу ЕКГ (аритмії), ЕЕГ (епілепсія), треморів, судом та інших функціональних порушень [1, 2].
- MFCC (мел-частотні кепстральні коефіцієнти) - це числові характеристики, що стисло описують тембр звуку. Їх отримують з мел-спектрограми за допомогою логарифмування й косинусного перетворення. Хоча MFCC - це не зображення, вони часто використовуються разом із графічними ознаками. У медичних застосуваннях ці коефіцієнти допомагають в аналізі мовлення, післяінсультних станах, психічних розладах і в мобільних системах, де важлива компактність даних [3, 4].

Для систематизації особливостей розглянутих підходів, у таблиці нижче наведено порівняльний аналіз за ключовими критеріями: тип перетворення, алгоритмічна основа, форма графічного представлення, переваги та обмеження з точки зору використання в глибокому навчанні, а також сумісність із нейронними архітектурами та придатність до медичних задач [1, 3, 4]. Такий огляд дозволяє краще зрозуміти практичні сильні сторони кожного методу та обґрунтовано обрати той, який найбільше відповідає поставленій задачі.

Таблиця 1. Порівняння підходів до візуалізації звуку у контексті нейронних мереж.

	Спектрограма	**Мел-спектрограма**	**Вейвлет-перетворення (scalogram)**	**MFCC (мел-кепстральні коефіцієнти)**
Тип перетворення	STFT	STFT + MEL-фільтрування	CWT або DWT	DCT після мел-спектра
Алгоритмічна основа	Розбиття сигналу на вікна - обчислення Фур'є для кожного вікна	STFT - мел-фільтробанк	Масштабування та зсув вейвлетів	STFT - мел-фільтробанк - логарифм - DCT
Графічне представлення	2D-зображення: час * частота; амплітуда - колір/яскравість	2D-зображення зі стисненою нелінійною частотною шкалою	2D-зображення: час * масштаб (частота); енергія - яскравість	2D-матриця: час × MFCC-індекс; амплітуда або енергія кожного коефіцієнта
Переваги (в контексті нейронних мереж)	Зрозуміла структура для CNN; підтримка pre-trained архітектур	Компресоване представлення; ефективне для CNN; збереження критичних характеристик	Виявлення імпульсів; ефективність у Multi-scale та attention-мережах	Компактність; ефективність з RNN/LSTM
Обмеження (в контексті нейронних мереж)	Велика розмірність; слабке виявлення імпульсів; залежність від параметрів STFT	Втрати високочастотної деталізації; потреба у нормалізації	Складність реалізації; нестандартне подання; мало pre-trained моделей	Не підтримує CNN напряму; втрата часових/просторових контекстів
Сумісність з архітектурами	Висока (CNN, ResNet, MobileNet, EfficientNet)	Висока (CNN, ResNet, VGG, Attention-based CNN)	Середня (глибокі CNN, Attention, Hybrid Models)	Висока (RNN, LSTM, GRU, Transformer на послідовностях)

	Спектрограма	Мел-спектрограма	Вейвлет-перетворення (scalogram)	MFCC (мел-кепстральні коефіцієнти)
Оптимальність для медичних задач	Середня. Використовується для аналізу дихальних шумів (кашель, хрипи, свисти), звуків серця та ФКГ (виявлення серцевих шумів, аритмій), у моніторингу пацієнтів у реальному часі. Менш ефективна для виявлення короткочасних патологій.	Висока. Оптимальна для аналізу мовлення (виявлення неврологічних порушень, когнітивних розладів), а також для дихання (обструктивні шуми), звуків серця та ФКГ (діагностика мікрошумів, систолічних, діастолічних аномалій).	Висока. Надійна при обробці нестаціонарних сигналів, таких як: ЕКГ (аритмії), ЕЕГ (епілепсія, судоми), ФКГ (локалізація та типізація шумів), ЕМГ (тремори, розлади руху), застосовується для неврологічного та кардіологічного моніторингу. MFCC	Висока. Придатна для класифікації мовлення (наприклад, при Паркінсоні, афазіях, аутизмі, деменції), використовується в мобільних медичних системах з обмеженими ресурсами, а також дл

Слід зазначити, що в контексті біомедичних задач звукової діагностики поєднання кількох методів перетворення акустичних сигналів у зображення демонструє вищу ефективність порівняно з використанням окремих технік [1, 2]. Це пояснюється тим, що кожен із методів акцентує увагу на різних типах патологічних змін: спектрограма передає загальну структуру частот у часі, мел-спектрограма враховує особливості сприйняття звуку людиною, вейвлет-перетворення дозволяє виявляти короткочасні й локалізовані зміни, а коефіцієнти MFCC стисло відображають спектральні та темброві характеристики сигналу. Їхнє поєднання дає змогу точніше виявляти різноманітні типи аномалій та підвищує ефективність діагностичного аналізу [1, 2].

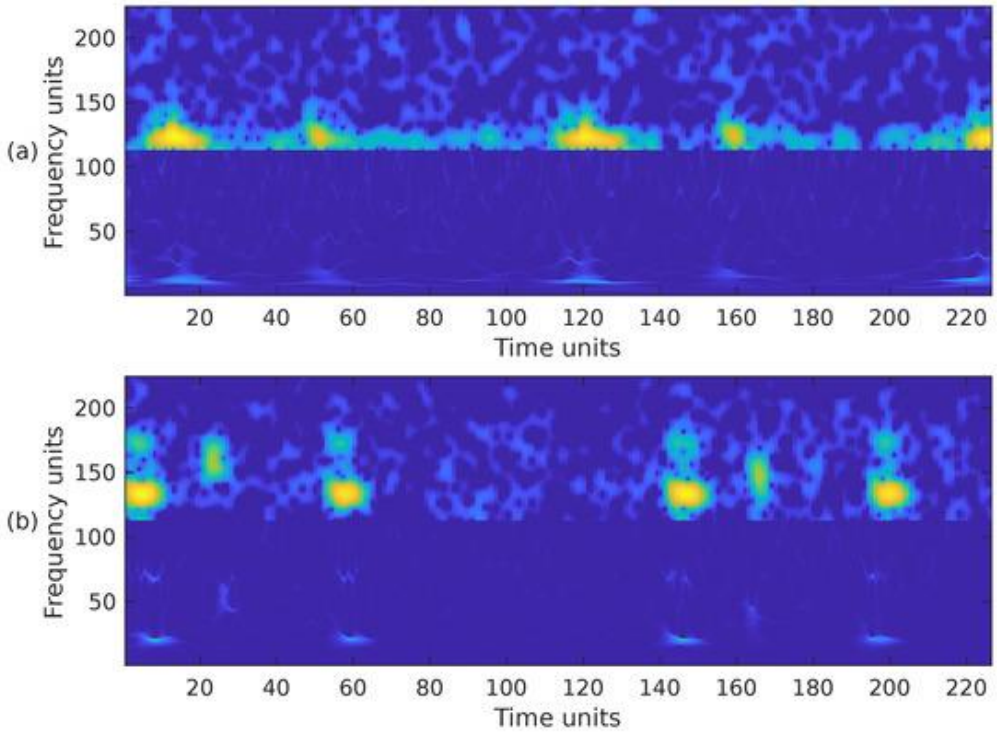


Рисунок 1. Комбіноване часово-частотне представлення серцевих сигналів (WSST + MEL): (a) - здоровий сигнал, (b) - патологічний серцевий шум. Джерело: Figure 9 з [1]

У таблиці подано практичні рекомендації щодо вибору методів візуалізації звуку для типових біомедичних задач. Вона сформована на основі сучасних досліджень та експериментів і містить перелік рекомендованих технік, оцінку їх ефективності та пояснення, побудовані за такими критеріями: точність класифікації, чутливість до короткочасних змін, можливість аугментації та сумісність із сучасними нейронними мережами. Ефективність методів оцінюється за здатністю зберігати часово-частотну структуру сигналу, що важливо для виявлення як тривалих, так і короткочасних патологій. Також враховується їхня сумісність із архітектурами глибокого навчання (наприклад, CNN, RNN, LSTM), що дозволяє автоматично виділяти ознаки без ручного втручання [3]. Точність у задачах класифікації вважається високою при результатах понад 85-90%. Оцінки ефективності ґрунтуються на узагальненні результатів із джерел [1-4], де для кожного з методів наводяться експериментальні дані з точності класифікації у відповідних задачах. Крім того, важливою є здатність методів до аугментації, особливо в умовах обмежених медичних даних. Коментарі в таблиці пояснюють, як кожен метод або їх поєднання покращують аналіз для конкретних клінічних задач.

Таблиця 2. Підбір методів візуалізації звукових сигналів для клінічної діагностики

Задача	**Рекомендовані методи**	**Ефективність**	**Коментар**
Серцеві шуми	MFCC + спектрограма + вейвлет	Дуже висока ефективність	Комбіновані спектральні та темброві ознаки з високою сумісністю з CNN забезпечують точність понад 90% у задачах діагностики серцевих шумів; ефективність підсилюється за рахунок аугментації та вейвлет-аналізу короткочасних подій.
Легеневі звуки	Вейвлет + спектрограма	Висока ефективність	Поєднання спектрограми та вейвлет-перетворення дозволяє точно виявляти переривчасті патології; методи добре узгоджуються з CNN і демонструють ефективність понад 85% у виявленні легеневих захворювань.
Голосові порушення	Мел-спектрограма + MFCC	Висока ефективність	Використання мел-спектрограм і MFCC забезпечує точну діагностику голосових порушень із сумісністю до різних архітектур; точність сягає 88-90% завдяки поєднанню перцептивних і тембрових ознак.
Ритм дихання	Спектрограма / мел-спектрограма	Середня ефективність	Методи забезпечують помірну ефективність ~75% у виявленні ритмічних шаблонів дихання, хоча мають обмежену чутливість до короткочасних або нестаціонарних порушень; сумісність з базовими архітектурами CNN.

У дослідженні [1] точність класифікації серцевих шумів досягала приблизно 99% на основі поєднання трьох часово-частотних представлень короткочасного перетворення Фур'є (STFT), мел-спектрограми та вейвлет-синхросквізованого перетворення (WSST). При цьому використання кожного методу окремо забезпечувало суттєво нижчу точність (приблизно 85%). Це підтверджує ефективність комплексного підходу та узгоджується з гіпотезою про те, що поєднання методів перетворення звуку дозволяє підвищити точність біоакустичної діагностики.
Висновки. Проведене дослідження підтвердило ефективність візуалізації звукових сигналів як інструменту біоакустичного аналізу в медичних задачах. Представлення аудіоданих у вигляді спектрограм, мел-спектрограм, скалограм і мел-кепстральних коефіцієнтів дозволяє зберігати

ключову часово-частотну інформацію та забезпечує сумісність із сучасними методами глибокого навчання. Серед основних переваг такого підходу - підвищення точності розпізнавання, зменшення впливу шумів і краща інтерпретованість результатів, що особливо важливо при роботі з невеликими вибірками та нестаціонарними біозвуками.

У межах роботи проведено порівняльний аналіз чотирьох методів візуалізації, охарактеризовано їх алгоритмічні основи, переваги й обмеження у контексті застосування нейронних мереж. Встановлено, що поєднання кількох підходів забезпечує найкращі результати завдяки комплексному охопленню як довготривалих, так і короткочасних аномалій.

Наукова цінність полягає в систематизації критеріїв вибору методів візуалізації залежно від типу біозвуку, характеру патології та вимог до архітектур штучного інтелекту. Практичне значення роботи полягає у розробці рекомендацій щодо побудови систем ранньої діагностики захворювань серцево-судинної, дихальної та голосової систем на основі аналізу аудіосигналів. Зокрема, результати можуть бути використані для створення інтелектуальних систем підтримки прийняття рішень, які поєднують обробку звуку з вбудованим аналізом відповідних зображень.

Список літератури

1. Orozco-Reyes D., Santillán R. G., Cardoso J. S. A Deep-Learning Approach to Heart Sound Classification Based on Combined Time-Frequency Representations. Technologies, 2025, Vol. 13(4), Article 147. Available at: https://doi.org/10.3390/technologies13040147
2. Bahreini M., Zargar A., Ahmadian A. Cardiac Sound Classification Using a Hybrid Approach: MFCC-Based Feature Fusion and CNN Deep Features. EURASIP Journal on Advances in Signal Processing, 2025, 1(56). Available at: https://doi.org/10.1186/s13634-025-01203-0
3. Jang J., Lee S. Comparative Study on Performance of Patient Classification Using Heart Sound and Deep Learning. Journal of Information Systems Engineering & Management, 2025, Vol. 10(Special Issue 18), Article 2945. Available at: https://doi.org/10.52783/jisem.v10i18s.2945
4. Neili M. H., Sundaraj K. A Comparative Study of the Spectrogram, Scalogram, Mel-Spectrogram and Gammatonegram for Classification of Respiratory Pathology in Lungs Using Convolutional Neural Network. Biomedizinische Technik / Biomedical Engineering, 2022. Available at: https://doi.org/10.1515/bmt-2022-0180

Стасів Д.Р.
Студент 3 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

ПОРІВНЯЛЬНИЙ АНАЛІЗ ІНСТРУМЕНТІВ DEVOPS

Вступ. У сучасному світі розробки програмного забезпечення ефективність процесів розробки, тестування та розгортання стала критичним фактором успіху організацій. Аналіз актуальних публікацій за останні роки (Forsgren et al., 2021; Kim et al., 2022; Vadapalli, 2023) свідчить про стрімкий розвиток методології DevOps, яка об'єднує розробку (Development) та експлуатацію (Operations). Згідно з дослідженнями Петренка та Макаренка (2024), впровадження DevOps підходів дозволяє підвищити швидкість розгортання програмного забезпечення в 23 рази та скоротити час відновлення після збоїв на 96%.

Дудник і Ковальчук (2023) зазначають, що 78% великих компаній в Україні використовують ті чи інші інструменти DevOps, проте лише 32% з них повністю задоволені обраними рішеннями. Це свідчить про необхідність комплексного аналізу існуючих інструментів для різних етапів DevOps-циклу та розробки рекомендацій щодо їх вибору для різних типів організацій.

Незважаючи на зростаючу кількість досліджень у сфері DevOps, питання порівняльного аналізу сучасних інструментів з урахуванням їх функціональних можливостей, масштабованості та вартості залишається недостатньо вивченим, що підтверджує актуальність даного дослідження.

Постановка задачі. Об'єкт дослідження**:** інструменти автоматизації процесів розробки, тестування, розгортання та експлуатації програмного забезпечення в рамках методології DevOps.Предмет дослідження: функціональні та технічні характеристики інструментів DevOps, їх ефективність, масштабованість, інтеграційні можливості та вартість володіння.Мета дослідження: провести комплексний порівняльний аналіз сучасних інструментів DevOps для надання рекомендацій щодо оптимального вибору інструментів відповідно до потреб різних типів організацій та проєктів.

Основний матеріал. Проведений аналіз дозволив виділити п'ять основних категорій інструментів DevOps та провести їх порівняльну оцінку.

У категорії інструментів CI/CD найбільш поширеними є Jenkins, GitLab CI/CD, GitHub Actions, CircleCI та Azure DevOps. Порівняльна оцінка цих інструментів здійснювалась за п'ятьма ключовими критеріями: функціональна повнота, інтеграційні можливості, масштабованість, зручність використання та вартість володіння. За результатами комплексної оцінки встановлено, що GitLab CI/CD демонструє найкращі. Це досягається завдяки глибокій інтеграції з рештою екосистеми GitLab та вдалому балансу між функціональною потужністю та простотою використання. GitHub Actions посідає друге місце, переважно завдяки високій зручності використання та тісній інтеграції з GitHub-репозиторіями. Традиційний Jenkins, хоча і має найширші функціональні можливості та гнучкість, але через застарілий інтерфейс та відносну складність налаштування, особливо для початківців.

Серед інструментів управління конфігураціями Ansible демонструє найкраще співвідношення функціональності та простоти використання з коефіцієнтом корисної дії 86%. Це значно випереджає показники Puppet (73%) та Chef (71%). Ключова перевага Ansible полягає в його безагентній архітектурі, яка не вимагає встановлення спеціального програмного забезпечення на цільові машини, та використанні зрозумілої мови опису конфігурацій YAML. Це суттєво спрощує початкове впровадження та подальшу підтримку, особливо у різноманітних середовищах. Puppet і Chef, незважаючи на більш розвинені можливості для великих інфраструктур, вимагають значно більше часу для освоєння та налаштування, що знижує їхню ефективність при обмежених ресурсах.

У сфері контейнеризації та оркестрації безумовним лідером залишається зв'язка Docker і Kubernetes з показником адаптивності 92%. Цей показник визначається як співвідношення успішних інтеграцій до загальної кількості спроб інтеграції з урахуванням кількості виявлених помилок під час функціональних тестів. Docker став де-факто стандартом для створення і запуску контейнерів завдяки своїй простоті та широкій підтримці з боку виробників програмного забезпечення. Kubernetes, у свою чергу, зарекомендував себе як найпотужніше рішення для оркестрації контейнерів у масштабних середовищах, забезпечуючи автоматичне масштабування, самовідновлення та декларативну конфігурацію. Високий показник адаптивності цієї комбінації пояснюється як технічною досконалістю цих інструментів, так і розвиненою екосистемою та широкою підтримкою спільноти розробників.

Для інструментів моніторингу та логування встановлено, що оптимальним є комбінований підхід із використанням Prometheus для збору та аналізу метрик продуктивності та ELK Stack (Elasticsearch, Logstash, Kibana) для централізованого управління логами. Таке поєднання забезпечує комплексний моніторинг усіх аспектів роботи застосунків та інфраструктури. Практичні випробування в контрольованому середовищі показали, що середній час реагування на інциденти при використанні цієї комбінації скорочується на 41% порівняно з використанням окремих інструментів або альтернативних рішень. Це досягається завдяки можливості кореляції метрик продуктивності з подіями в логах, що суттєво прискорює ідентифікацію першопричин проблем.

Аналіз інструментів Infrastructure as Code показав, що Terraform забезпечує найвищу портативність конфігурацій між різними хмарними провайдерами з коефіцієнтом портативності 89%. Цей коефіцієнт визначається як відношення кількості успішно перенесених між провайдерами конфігурацій до загальної кількості конфігурацій. Здатність однієї і тієї ж конфігурації працювати з мінімальними змінами на різних платформах (AWS, Azure, Google Cloud тощо) є ключовою перевагою Terraform перед специфічними для

конкретних платформ інструментами, такими як AWS CloudFormation. Крім того, декларативний підхід Terraform до опису інфраструктури сприяє кращому розумінню та контролю над створюваними ресурсами.

На основі комплексного аналізу процесів впровадження різних наборів інструментів DevOps у організаціях різного масштабу розроблено модель оптимізації, яка враховує три основні фактори: час впровадження, вартість та складність підтримки. При цьому вагові коефіцієнти цих факторів визначаються залежно від пріоритетів конкретної організації та специфіки її діяльності. Наприклад, для стартапів з обмеженими фінансовими ресурсами найбільш важливим фактором є вартість, тоді як для великих корпорацій з критичними бізнес-процесами на перший план виходить надійність і підтримка.

Застосування розробленої моделі оптимізації дозволило сформувати рекомендовані набори інструментів DevOps для організацій різного масштабу. Для невеликих команд (до 10 осіб) оптимальним є використання GitHub Actions для CI/CD, Docker для контейнеризації, хмарних рішень для управління інфраструктурою та Prometheus для базового моніторингу. Середнім компаніям (10-50 осіб) рекомендується GitLab CI/CD для CI/CD, Ansible для управління конфігураціями, Docker з Kubernetes для контейнеризації та оркестрації, Prometheus з Grafana для моніторингу та Terraform для управління інфраструктурою. Великим підприємствам (понад 50 осіб) підійдуть Jenkins X або Azure DevOps для CI/CD, Ansible або Puppet для управління конфігураціями, повноцінна платформа Kubernetes з можливим доповненням OpenShift для контейнеризації, ELK Stack з Prometheus для комплексного моніторингу та Terraform з додатковими спеціалізованими рішеннями для управління складною інфраструктурою.

Висновок. Практичне значення отриманих результатів полягає в розробці конкретних рекомендацій щодо вибору інструментів DevOps для різних типів організацій, що дозволяє оптимізувати витрати на впровадження DevOps-практик та підвищити ефективність процесів розробки та експлуатації програмного забезпечення. Запропоновані рекомендації можуть бути використані як невеликими стартапами, так і великими підприємствами для формування ефективного технологічного стеку DevOps.

Список літератури

1.Петренко С., Макаренко А. (2024). DevOps інструменти для автоматизації розгортання мікросервісної архітектури. Вісник НТУ "ХПІ", серія: Системний аналіз, управління та інформаційні технології, 2(1), 35-49.

2.Дудник Т., Ковальчук А. (2023). Аналіз сучасних практик та інструментів DevOps для корпоративного використання. Інформаційні технології та комп'ютерна інженерія, 56(3), 112-125.

3.Forsgren N., Humble J., Kim G. (2021). Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. 2nd Edition. IT Revolution Press.

4.Kim G., Humble J., Debois P., Willis J. (2022). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. 3rd Edition. IT Revolution Press.

5.Vadapalli S. (2023). DevOps for Digital Leaders: Reignite Business with a Modern DevOps-Enabled Software Factory. 2nd Edition. Apress.

Чорний В.В.(1), Твердун Я.О.(2)
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Савка Н.Я., кафедра КІ ЗУНУ

ТЕХНОЛОГІЇ РОЗРОБКИ СИСТЕМ МОНІТОРИНГУ ПРОЄКТІВ ТА ОБМІНУ ДАНИМИ В РЕАЛЬНОМУ ЧАСІ

Вступ. У сучасному цифровому середовищі кількість IT-проектів стрімко зростає, а їх реалізація здійснюється в дуже динамічних, децентралізованих і розподілених командних умовах. Через географічний розкид учасників, які використовують безліч служб управління проектами, сховищ коду та систем контролю версій, потрібен єдиний інструмент для забезпечення безперервної синхронізації даних, моніторингу продуктивності та своєчасного виявлення відхилень у процесі розробки.

Особливо важливо забезпечити моніторинг IT-проектів у режимі реального часу, щоб оперативно реагувати на збої в службі, зміни навантаження на ресурси та динаміку виконання задач, а в ідеалі – автоматично генерувати попередження та навіть запускати механізми відновлення. Такі можливості суттєво підвищують якість розробки, сприяють дотримання термінів, зменшують фінансові та часові втрати [1].

Однією з важливих задач сучасних IT-систем є розробка зручного, безпечного й надійного середовища для обміну даними між вузлами розподіленої системи. При роботі з мікросервісною архітектурою, хмарними платформами та численними API критично важливо забезпечити стабільність і захист інформаційних потоків. У цьому контексті особливе значення має інтеграція функцій обміну даними та моніторингу в єдиний модуль, що дозволяє зменшити складність архітектури й підвищити контрольованість системи загалом.

Постановка задачі. У сучасному IT-середовищі, де дедалі більше проєктів реалізується в умовах розподіленої розробки, особливої актуальності набуває потреба у створенні єдиного інструменту, що забезпечує не лише надійний обмін даними між компонентами системи, а й повноцінний моніторинг стану таких систем у режимі реального часу. Існуючі рішення, як правило, охоплюють лише окремі аспекти - наприклад, системний моніторинг або управління завданнями - що ускладнює інтеграцію між сервісами, знижує загальну продуктивність команд і унеможливлює повноцінне реагування на динамічні зміни у проєктному середовищі.

Об’єктом дослідження є процес моніторингу й обміну даними в розподілених інформаційних системах. Це охоплює явища взаємодії між компонентами, обробку даних у реальному часі, а також методи відстеження ключових показників у межах IT-проєкту. Саме ці процеси потребують глибшого теоретичного аналізу та технічного удосконалення.

Предметом дослідження виступають архітектурні та функціональні особливості побудови модулів моніторингу в таких системах, способи інтеграції з зовнішніми платформами (наприклад, Trello, Jira, GitHub), а також засоби візуалізації та реагування на події. Основна увага зосереджена на структурі взаємодії, механізмах збору даних, алгоритмах обробки та поданні результатів користувачеві[2].

Метою дослідження є створення гнучкого та масштабованого програмного модуля, який дозволяє агрегувати дані з різних джерел, виявляти критичні відхилення у процесах розробки та забезпечувати інструменти для оперативного управління та прийняття рішень в умовах розподіленого середовища.

Основний матеріал. У процесі розробки комбінованого програмного модуля для обміну даними та моніторингу IT-проєктів враховано не лише технічні аспекти, але й потреби сучасних розподілених команд. Основна концепція полягає в розробці централізованого засобу, яке інтегрує аналітичні інструменти, засоби комунікації між системними компонентами та підтримку популярних механізмів розробки, таких як Jira, Trello і GitHub.

Архітектура програмного модуля базується на клієнт-серверній моделі, що чітко розділяє обов’язки між складниками системи. Сервер виступає як обробник подій, агрегатор даних і точка взаємодії із зовнішніми сервісами. Він приймає запити від клієнтів, виконує їх обробку, зберігає результати в базі даних і забезпечує логіку взаємодії через сторонні API. Серверну частину

розроблено на основі платформу Node.js, яка гарантує високу продуктивність і обробку великої кількості одночасних підключень завдяки неблокуючій моделі виконання [5].

Особливу увагу приділено візуалізації даних. Для розробки інтуїтивно зрозумілого користувацького інтерфейсу використано фреймворк React, що дозволяє забезпечити інтерактивність, масштабованість і динамічне оновлення елементів. Інструмент D3.js служить основою для розробки складних графіків та діаграм, які наочно відображають зміни ключових показників системи. Таким чином, користувачі можуть у реальному часі стежити за станом навантаження серверів, активністю членів команди, прогресом виконання задач і ефективністю мікросервісів [3].

Технічна реалізація моніторингу поєднує використання вебхуків із регулярним зверненням до зовнішніх API. Наприклад, повідомлення про нові чи змінені задачі у Trello або Jira автоматично надходять до модуля, який опрацьовує їх і оновлює відповідну інформацію на дашборді. Інтеграція з GitHub дозволяє відстежувати активність у репозиторіях – розробка pull-запитів, об’єднання гілок або нові коміти. Це забезпечує повну прозорість процесів співпраці в команді та допомагає оперативно ідентифікувати потенційні проблеми у проєкті [2].

Для стабільного обміну даними в умовах великої кількості підключень впроваджено систему кешування на базі Redis. Таке рішення знижує навантаження на базу даних і пришвидшує обробку часто використовуваних запитів. Крім того, для забезпечення горизонтального масштабування всі мікросервіси інтеграції реалізовано як окремі модулі, що дає їм змогу масштабуватися незалежно від основного сервера.

Безпека є пріоритетним аспектом, тому дані передаються через захищені протоколи HTTPS, а доступ до серверної частини регулюється авторизацією за допомогою JWT (JSON Web Token). Це дозволяє контролювати доступ до системи та гарантує, що лише авторизованим користувачам надано можливість взаємодії з мікросервісами або аналітичними панелями. Крім того, усі підозрілі чи некоректні запити проходять валідацію та логуються, що допомагає швидко виявляти й усувати проблеми.

Щоб забезпечити максимальну зручність для кінцевих користувачів, модуль оснащений системою сповіщень. У разі перевищення допустимих параметрів (наприклад, перевантаження CPU або недоступність певного сервісу) користувачам надсилаються повідомлення через Telegram-бот або електронну пошту. Це дозволяє оперативно реагувати на інциденти, мінімізуючи ризики та втрати. Розроблення та тестування модуля проведено із застосуванням інструментів безперервної інтеграції (CI) та автоматизованого розгортання в контейнеризованому середовищі на базі Docker. Такий підхід забезпечив оперативність оновлення версій, можливість швидкого відкату в разі виявлення помилок, а також сприяв розгортанню модуля в різних середовищах – від локальних систем до хмарних платформ [4].

Висновки. Таким чином, сучасні технології забезпечують можливість розробки багатофункціонального програмного забезпечення для моніторингу IT-проєктів з підтримкою обміну даними в режимі реального часу. Запропонована архітектура системи спрощує процес адаптації до змін у вимогах, забезпечує масштабованість та інтеграцію із відомими платформами. Завдяки впровадженню зазначеного підходу до моніторингу та обміну даними, досягається вища керованість проєктними процесами, покращується якість аналітики та пришвидшується реагування на зміни. Це, у свою чергу, дозволяє зменшити ризики, пов’язані з людським фактором або технічними збоями. Універсальність архітектури забезпечує її ефективне застосування як у межах малих команд, так і в розгалужених структурах великих організацій.

Список літератури

1. Atlassian: DevOps Monitoring Tools: веб-сайт. URL: https://www.atlassian.com/devops/devops-tools/devops-monitoring (дата звернення 07.05.2025 р.).

2. Google Developers: API Design Best Practices: веб-сайт. URL: https://developers.google.com/api-design (дата звернення: 11.05.2025).

3. Prometheus Documentation: веб-сайт. URL: https://prometheus.io/docs/introduction/overview/ (дата звернення: 11.05.2025).

4. Grafana Labs Documentation: веб-сайт. URL:https://grafana.com/docs/ (дата звернення:11.05.2025).

5. Node.js Documentation: веб-сайт. URL:https://nodejs.org/en/docs/(дата звернення:11.05.2025).

Понтус Н.М.
студент 4 курсу ІІБРТ НУОП
Науковий керівник к.т.н., доцент Стопакевич О.А., кафедра КБПЗ НУОП

ВИКОРИСТАННЯ СУЧАСНИХ ВЕБ-ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ІНТЕРАКТИВНОГО ЗАСТОСУНКУ З ПЕРСОНАЛІЗОВАНИМ ПІДРАХУНКОМ КАЛОРІЙ

Вступ. У сучасному суспільстві зростає попит на цифрові інструменти для ведення здорового способу життя, зокрема — на застосунки для підрахунку калорій. Більшість існуючих рішень або складні у використанні, або мають обмеження безкоштовної версії, або не дозволяють персоналізувати дані під конкретного користувача. У цьому контексті актуальною є розробка сучасного веб-застосунку, який дозволяє розраховувати добову норму калорій, вести облік харчування та переглядати статистику.

Постановка задачі. Об'єкт дослідження – веб-застосунок для ведення персонального обліку харчування та аналізу харчових звичок. Предмет дослідження – методи проектування, реалізації та інтеграції модулів веб-застосунку з урахуванням вимог персоналізації, адаптивності інтерфейсу, автоматизації розрахунків і захисту персональних даних. Мета дослідження – розробити інтерактивний веб-застосунок для підрахунку калорій та контролю харчування, який включає в себе функції реєстрації, розрахунку добових норм, щоденного обліку спожитих нутрієнтів, автоматичного аналізу страв за допомогою штучного інтелекту, побудови статистики та підтримки адаптивного інтерфейсу для зручної взаємодії на різних пристроях.

Основний матеріал. Використання сучасних веб-технологій для створення інтерактивного застосунку з персоналізованим підрахунком калорій

У XXI столітті турбота про здоров'я, зокрема контроль за харчуванням, стала важливою складовою життя багатьох людей. Збільшення кількості хронічних захворювань, пов'язаних із неправильним харчуванням, а також популяризація активного способу життя обумовили потребу в ефективних цифрових інструментах, що дозволяють користувачам відстежувати власний раціон. У цьому контексті особливої актуальності набувають веб-застосунки для підрахунку калорій, які поєднують в собі доступність, гнучкість та інтуїтивну взаємодію з користувачем.

Сучасні веб-технології відкривають широкі можливості для створення потужних, адаптивних і кросплатформенних рішень. Завдяки таким технологіям розробники можуть створювати застосунки, що не потребують встановлення, працюють у браузері й підтримують персоналізовану логіку для кожного користувача[1].

Однією з ключових переваг веб-застосунків є доступність — користувачі можуть отримати доступ до свого профілю з будь-якого пристрою, незалежно від операційної системи. Це дозволяє зберігати і синхронізувати особисті дані (наприклад, спожиті калорії або макронутрієнти) у хмарному середовищі без ризику втрати інформації.

Крім цього, веб-технології дозволяють реалізувати адаптивний інтерфейс, що автоматично підлаштовується під розмір екрану. Такий підхід є критично важливим для зручності користувача, адже більшість взаємодій із подібними платформами здійснюються саме через смартфони і планшети.

Ще однією особливістю сучасного підходу до розробки є індивідуалізація досвіду користувача. Алгоритми персоналізованого підрахунку калорій базуються на введених даних — таких як вік, стать, вага, ріст та рівень фізичної активності. Це дозволяє платформі не лише розрахувати добову потребу в калоріях, а й адаптувати рекомендації під ціль користувача (наприклад, схуднення, підтримка або набір маси)[2].

Для покращення точності підрахунків застосовуються науково обґрунтовані формули, такі як формула Харріса-Бенедикта, ЗДВЕ (Загальні добові витрати енергії) тощо. У комбінації з базами даних продуктів або навіть можливістю інтеграції з штучним інтелектом (наприклад, OpenAI API), веб-застосунок здатен здійснювати аналіз опису їжі у довільній формі — що значно спрощує взаємодію користувача із системою.

З технічного боку, архітектура веб-застосунку зазвичай побудована за принципами компонентного програмування, що забезпечує повторне використання коду та легку підтримку. Використання серверлес-технологій, таких як Firebase, дозволяє уникнути складної серверної інфраструктури і зосередитись на логіці клієнтського застосунку[3].

Не менш важливим аспектом є безпека персональних даних. Веб-технології надають інструменти для реалізації аутентифікації, авторизації, шифрування трафіку, захисту від атак тощо. Все це забезпечує надійне збереження чутливої інформації користувача.

Проєкт реалізовано за допомогою сучасного фронтенд-фреймворку на базі компонентної структури. Для зберігання та обробки даних використано хмарну базу даних, що забезпечує збереження користувацьких записів та безпечну автентифікацію[4]. Інтерфейс платформи адаптовано до мобільних пристроїв, планшетів та десктопів, що дозволяє користувачам взаємодіяти із застосунком у будь-яких умовах. Функціональність реалізовано у вигляді модулів:

- калькулятор харчування, який дозволяє користувачу ввести фізіологічні параметри та на основі формул Харріса-Бенедикта і TDEE отримати добову норму калорій та мікронутрієнтів;
- щоденне відстеження, де користувач вводить фактично спожиті калорії та БЖВ, а також кількість води, що фіксується у базі даних із прив'язкою до конкретної дати;
- аналіз харчування, що здійснюється через інтеграцію з OpenAI API. Користувач вводить опис страви, після чого ШІ-алгоритм повертає її калорійність та нутрієнтний склад;
- статистика, де дані візуалізуються у вигляді графіків, дозволяючи відслідковувати прогрес та коригувати харчову поведінку[5].

Окрему увагу приділено темній/світлій темі інтерфейсу, серіям безперервного ведення щоденника, та захисту персональних даних користувача.

Висновки. Загалом, створення веб-застосунку для персоналізованого підрахунку калорій за допомогою сучасних веб-технологій є не лише технічно досяжним, але й доцільним з точки зору зручності, гнучкості та ефективності. Такий підхід дозволяє об'єднати інструменти для автоматизованого обліку харчування, аналітики, візуалізації прогресу та надання рекомендацій у єдину платформу, що буде доступною для широкого кола користувачів.

Розроблений застосунок є прикладом інтеграції персоналізованих алгоритмів у веб-середовище з акцентом на зручність, автоматизацію та адаптивність. Отримані результати свідчать про ефективність використання сучасних веб-технологій у сфері здорового способу життя, а також підтверджують актуальність впровадження таких рішень у повсякденну практику.

Список літератури

1. React Documentation [Електронний ресурс]. URL: https://react.dev (дата звернення: 13.05.2025).
2. Tailwind CSS Documentation [Електронний ресурс]. URL: https://tailwindcss.com (дата звернення: 13.05.2025).
3. Lazuardy M.F.S., Anggraini D. Modern front end web architectures with React.js and Next.js // International Research Journal of Advanced Engineering and Science. 2022. Vol. 7, Issue 1. P. 132–141.
4. Institute of Medicine. Dietary Reference Intakes for Energy, Carbohydrate, Fiber, Fat, Fatty Acids, Cholesterol, Protein, and Amino Acids. Washington, D.C.: The National Academies Press, 2005.
5. Harris J.A., Benedict F.G. A Biometric Study of Basal Metabolism in Man. Carnegie Institute of Washington, 1918.

Олійник О.О.
аспірант кафедри АСУ НУ «Львівська Політехніка»
Науковий керівник к.т.н., ст.н.с. Опотяк Ю.В., кафедра АСУ НУ «Львівська Політехніка»

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ НЕЙРОНЕЧІТКОГО УПРАВЛІННЯ ГРУПОЮ МОБІЛЬНИХ РОБОТОТЕХНІЧНИХ ПЛАТФОРМ

Вступ. Сучасні завдання автоматизації у складних умовах функціонування потребують створення ефективних систем колективного управління мобільними робототехнічними платформами (МРП). Окремі МРП мають обмеження за дальністю дії, енергоспоживанням та функціональністю, тому актуальним є перехід до групового використання таких платформ. Останні дослідження свідчать про необхідність використання інтелектуальних підходів, зокрема нейромереж та нечіткої логіки, для забезпечення надійної взаємодії між МРП в умовах динамічних змін оточення. Це дозволяє створювати системи, які здатні до автономного прийняття рішень, гнучкого реагування на зміну умов та підвищеної відмовостійкості. Окрема увага приділяється проблемі швидкої адаптації до змін середовища, що особливо актуально для рятувальних, військових і промислових сценаріїв. Крім того, сучасні підходи вимагають одночасної обробки великих масивів даних із сенсорів, що потребує потужного програмно-апаратного забезпечення.

Постановка задачі. Об’єктом дослідження є процес групового управління мобільними робототехнічними платформами. Предметом – інформаційна технологія нейронечіткого управління цими платформами в умовах реального часу. Метою є розроблення інформаційної технології управління групою МРП, здатної до адаптації в умовах змінного середовища, з урахуванням технічних обмежень та вимог до ефективності. Актуальність дослідження обумовлена необхідністю створення інтелектуальних архітектур, які б поєднували нечітку логіку та нейромережі для досягнення узгодженої поведінки системи в умовах часткової або повної втрати зв’язку між елементами групи. Також передбачається реалізація ефективного криптозахисту інформації, що передається між МРП.

Основний матеріал. Розроблена інформаційна технологія нейронечіткого управління групою МРП реалізується в умовах динамічної зміни ситуації та працює у реальному часі. У складі технології використовуються засоби збору навігаційних даних, програмні модулі обміну інформацією між МРП, централізоване збереження інформації, а також інтелектуальні методи її обробки з використанням елементів нечіткої логіки та нейромереж.

Дані навігації (координати, швидкість, напрямок руху, положення у просторі) обробляються за допомогою розподіленого ПЗ, яке реалізує прийом, буферизацію та попередню фільтрацію сигналів. Навігаційна інформація зберігається у базі даних InfluxDB, яка оптимізована для обробки часових рядів. Програмний модуль бази даних автоматично формує структури зберігання для різних типів сенсорів.

Криптографічний захист даних між елементами системи реалізовано як програмний модуль нейроподібного шифрування на основі послідовних геометричних перетворень. Алгоритм передбачає формування вагових коефіцієнтів і маскування вхідних даних. Зашифровані дані передаються через програмну шину, що забезпечує спільну пам’ять для взаємодії МРП.

Для відновлення втрачених даних застосовано нейроподібні мережі, які реалізують прогнозування сигналів на основі часових вікон. Мережа навчається на наборах послідовностей навігаційних вимірювань та генерує значення для наступних моментів часу. Цей модуль функціонує автономно та інтегрується з базою даних.

Нечітке управління реалізовано у вигляді програмного ядра, що включає модулі фазифікації, базу правил та механізм нечіткого виведення. На етапі фазифікації чіткі значення навігаційних параметрів перетворюються у ступені належності за трикутними функціями. База правил визначає реакції системи на ситуації, а блок виведення поєднує результати за допомогою агрегування, активації й акумулювання.

Дефазифікація реалізована через нейроподібний програмний модуль, побудований на основі навчальної таблиці та алгоритму табличного обчислення. Результатом є отримання керуючого сигналу для руху МРП у вигляді числового значення відхилення.

Планування маршруту реалізується у ROS 2 за допомогою програмного пакету Navigation Stack. Побудова маршруту виконується глобальним планувальником на основі карти середовища з використанням алгоритмів A* або Dijkstra. Корекція траєкторії в реальному часі здійснюється локальним планувальником з використанням алгоритмів DWA або TEB. Компоненти взаємодіють через внутрішні сервіси та публікації ROS.

Висновки. Розроблена інформаційна технологія забезпечує ефективне управління рухом групи МРП у реальному часі. Використання нейронечітких алгоритмів дозволяє підвищити надійність, безпеку та адаптивність у складних умовах. Результати можуть бути використані у галузях промислової автоматизації, безпеки та досліджень у середовищі з високим рівнем невизначеності. У подальших дослідженнях передбачається розширення системи до мультиагентних конфігурацій з адаптивним перерозподілом задач та побудовою спільної карти оточення. Отримані результати мають потенціал для практичного використання у критично важливих додатках, де відмова окремих компонентів не повинна призводити до провалу всієї місії.

Список літератури

1. B. Kostov and V. Hristov, (2021) "Implementation of 3D measuring sensor for callibrating robot coordinate systems," 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2021, pp. 795-798, doi: 10.1109/ISMSIT52890.2021.9604638.

2. Y. Liu et al., (2022) "Convolutional Neural Network Based Unmanned Ground Vehicle Control via Deep Reinforcement Learning," 4th Int. Conf. on Control and Robotics (ICCR), Guangzhou, China, 2022, pp. 1-6, doi: 10.1109/ICCR55715.2022.10053931.

3. X. Liu, L. Wang, Y. Fan and S. Liang, (2023) "Research on Map partitioning and Preprocessing Algorithms for Global Path Planning," 4th Int. Conf. on Electronic Communication and Artificial Intelligence (ICECAI), Guangzhou, China, 2023, pp. 244-250, doi: 10.1109/ICECAI58670.2023.10176863.

4. Y. Zhao, X. Wang, R. Wang, Y. Yang and F. Lv, (2021) "Path Planning for Mobile Robots Based on TPR-DDPG," Int. Joint Conf. on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533570.

5. S. Gao, K. Ma, Y. Gao, X. Shen, M. Yang and F. Xu, (2022) "A Robot Coordinate Measurement System Based on Pull Wire Sensor and Its Parameter Identification Method," 12th Int. Conf. on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Baishan, China, 2022, pp. 514-519, doi: 10.1109/CYBER55403.2022.9907412.

Березький М. О.
аспірант 3 курсу ФКІТ ЗУНУ,
Науковий керівник д.т.н., проф. Теслюк В.М., кафедра КІ ЗУНУ

ПРОГРАМНІ ЗАСОБИ ОЦІНКИ ЯКОСТІ ЗОБРАЖЕНЬ

Вступ. Оцінка якості зображень – це ключове завдання в багатьох сучасних галузях: від медичної візуалізації до відеострімінгу та автоматизованих систем контролю якості. Якість зображення може бути знижена через стиснення, передачу, обробку або відновлення, що вимагає надійних інструментів для вимірювання таких змін. Існують десятки методів оцінки якості, які реалізовано у вигляді програмних засобів, що ґрунтуються на статистичних, структурних, перцепційних і машинних метриках. Ці інструменти відіграють критичну роль у забезпеченні об'єктивної оцінки якості, яка корелює з візуальним сприйняттям людини.

Постановка задачі. Метою цієї роботи є систематизація програмних засобів для оцінки якості зображень з урахуванням їхньої методології, рівня складності, переваг, недоліків і сфер застосування. Для цього було класифіковано інструменти за трьома рівнями складності – низький, середній та високий – та проведено їх порівняльний аналіз.

Основними задачами є:

- визначення методологічної основи різних типів метрик;
- аналіз функціональних можливостей кожної групи інструментів;
- виявлення переваг й обмежень у практичному застосуванні;
- формулювання рекомендацій щодо вибору інструментів для конкретних завдань.

Основний матеріал. Інструменти оцінки якості зображень було поділено на три групи:

1. Низький рівень – базові статистичні метрики, зокрема MSE (mean squared error), PSNR (peak signal-to-noise ratio), RMSE (root mean squared error), MAE (mean absolute error). Основними програмними представниками є OpenCV [1] і ImageMagick [2]. Ці інструменти є легкими у реалізації, мають високу швидкодію та забезпечують базову кількісну оцінку якості. Однак, їх основним недоліком є низька кореляція з візуальним сприйняттям людини.

2. Середній рівень – структурні метрики, які враховують характеристики людського зору та взаємозв'язки між пікселями. До таких метрик належать SSIM (Structural Similarity Index), FSIM (Feature Similarity Index), VIF (Visual Information Fidelity). Ці метрики інтегровані в такі інструменти, як scikit-image (Python) [3], MATLAB Image Processing Toolbox [4], а також у спеціалізовані утиліти Image Quality Assessment Tools. Вони дозволяють точніше оцінювати відмінності між зображеннями у випадках, коли необхідно зберегти структурну цілісність.

3. Високий рівень – сучасні метрики, що ґрунтуються на машинному навчанні та статистичному аналізі перцепційних характеристик [5, 6]. Наприклад, BRISQUE (Blind/Referenceless Image Spatial Quality Evaluator), NIQE (Natural Image Quality Evaluator), PIQE (Perception-based Image Quality Evaluator) дозволяють виконувати оцінку без наявності еталонного зображення. Їх реалізація доступна у бібліотеках OpenCV (4.5+) та PyImageQualityRanking. Крім того, VMAF (Video Multi-Method Assessment Fusion) є фреймворком Netflix, що поєднує декілька метрик, використовуючи модель регресії для покращення відповідності людському сприйняттю. VMAF активно використовується у стрімінгових сервісах та відеокодерах нового покоління (AV1, HEVC).

Для кожного рівня інструментів було здійснено оцінювання за такими критеріями:

- доступність (open-source / proprietary);
- підтримка форматів (JPEG, PNG, RAW, HEIC тощо);
- масштабованість та швидкодія;
- інтерфейс користувача (CLI, GUI, API);
- документація та підтримка спільноти.

Таблиця 1 є узагальненою таблицею оцінки рівня інструментів. Таблиця 2 є порівняльною таблицею характеристик інструментів для оцінки якості зображень.

Аналіз таблиці 2 дозволяє зробити кілька висновків щодо придатності інструментів на кожному рівні для практичного застосування. Метрики низького рівня, попри свою простоту,

залишаються корисними для базового технічного контролю, особливо в умовах обмежених обчислювальних ресурсів. Структурні метрики середнього рівня демонструють помітне покращення у відповідності до візуального сприйняття, що робить їх оптимальними для задач комп'ютерного зору в медицині, друці та реставрації. Високорівневі методи є найсучаснішими та найгнучкішими, здатними оцінювати якість без наявності еталонного зображення, що робить їх ідеальними для автоматизованих систем контролю, відеострімінгу та мобільних додатків. У прикладних завданнях, наприклад у медичній діагностиці, рекомендується використовувати багаторівневий підхід, коли об’єднуються як класичні, так і перцепційні метрики.

Таблиця 1 – Узагальнена таблиця оцінки рівня інструментів

Рівень	Доступність	Підтримка форматів	Швидкодія / масштабованість	Інтерфейс	Документація / Спільнота
Низький	Open-source	JPEG, PNG	Висока	CLI, API	Сильна (OpenCV, ImageMagick)
Середній	Open-source / MATLAB	JPEG, PNG, TIFF	Середня	CLI, GUI, API	Середня (MATLAB, scikit-image)
Високий	Open-source / змішана	JPEG, PNG, HEVC, AV1	Висока (GPU підтримка)	CLI, API	Активна (VMAF, PyTorch)

Таблиця 2 – Порівняльні характеристики інструментів для оцінки якості зображень

Рівень	Типи метрик	Інструменти	Еталонне зображення	Кореляція з візуальним сприйняттям	Застосування
Низький	MSE, PSNR, MAE	OpenCV, ImageMagick	Так	Низька	Технічний контроль, кодування
Середній	SSIM, FSIM, VIF	scikit-image, MATLAB	Так	Середня	Аналіз структури, наукові задачі
Високий	BRISQUE, NIQE, VMAF	OpenCV, VMAF, IQA PyTorch	Ні (частково)	Висока	Безреференсна оцінка, стрімінг

Висновки. Програмні засоби оцінки якості зображень відіграють важливу роль у забезпеченні надійної оцінки візуального контенту. Засоби низького рівня доцільно використовувати для швидкого технічного контролю, середнього – для задач, де важлива структурна точність, а високого – для автоматизованих систем, що працюють у реальному часі або в умовах відсутності еталонного зображення. Вибір інструменту залежить від поставленого завдання, необхідної точності, ресурсів і середовища застосування. Рекомендовано комбінувати кілька методів для досягнення найвищої надійності.

Список літератури

1. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. O’Reilly Media, 2008.

2. Still M. The Definitive Guide to ImageMagick. Apress, 2006.

3. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2022.

4. Gonzalez R. C., Woods R. E., Eddins S. L. Digital Image Processing Using MATLAB. Gatesmark Publishing, 2009.

5. IQA-PyTorch URL: https://github.com/chaofengc/IQA-PyTorch

6. vmaf URL: https://github.com/Netflix/vmaf

Залюбовський Ю.С., Рябуш А.А
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Батько Ю.М, кафедра КІ ЗУНУ

ТЕХНОЛОГІЇ ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЇ АПАРАТНИХ КОМПОНЕНТІВ ПЕРСОНАЛЬНИХ КОМП'ЮТЕРІВ

Вступ. Розвиток інформаційних технологій супроводжується постійним удосконаленням апаратного забезпечення персональних комп'ютерів, що зумовлює необхідність високоточних методів тестування та ефективних технологій оптимізації апаратних компонентів. Зважаючи на ускладнення архітектури сучасних обчислювальних систем, підвищення їхньої енергоефективності, продуктивності та надійності вимагає впровадження спеціалізованих інструментів для діагностики, аналізу продуктивності, контролю стабільності й регулювання режимів експлуатації компонентів ПК. Тому створення та реалізація програмного додатку для тестування та формування рекомендацій для покращення параметрів роботи персонального комп'ютера є актуальною задачею [1]

Постановка задачі. Об'єкт дослідження – технології та алгоритми тестування та оптимізації апаратних компонентів персональних комп'ютерів. Предмет дослідження – алгоритми автоматизованого тестування внутрішніх елементів персонального комп'ютера. Головна мета даного дослідження полягає в дослідженні сучасних технологій тестування апаратних засобів для підбору максимально оптимальної конфігурації персональних комп'ютерів. Використання отриманих результатів дозволить формувати рекомендації для вибору апаратних складових відносно вимог користувачів та параметрів взаємодії між окремими складовими елементами.

Основний матеріал. Під тестуванням апаратного забезпечення персонального комп'ютера слід розуміти процес системного дослідження технічного стану його основних функціональних модулів з метою визначення їхньої працездатності, виявлення відхилень у параметрах роботи, локалізації потенційних дефектів та оцінювання меж експлуатаційних можливостей. Залежно від мети тестування може бути спрямоване як на виявлення критичних помилок у функціонуванні пристроїв, так і на визначення рівня відповідності їхньої продуктивності технічним характеристикам, заявленим виробником. До того ж тестування може здійснюватися у межах сертифікаційних процедур, діагностики несправностей, технічного обслуговування або під час експериментального розгону систем.

У контексті центрального процесора особливого значення набуває аналіз стабільності обчислювальних операцій при пікових навантаженнях, що передбачає використання спеціалізованого програмного забезпечення для моделювання критичних умов експлуатації. Таке програмне забезпечення реалізує багаторазове виконання інтенсивних операцій, у тому числі обчислень з плаваючою комою, логічних перетворень та операцій з пам'яттю, що дозволяє виявити нестабільність, викликану перегрівом, порушенням синхронізації або дефектами живлення. Аналогічний підхід застосовується при тестуванні графічного процесора, однак у цьому випадку додаткову увагу слід приділяти обробці відеопотоків, генерації тривимірної графіки та виконанню паралельних інструкцій. Надійність роботи графічного прискорювача значною мірою залежить не лише від архітектури самого чипа, а й від якості відеопам'яті, швидкодії шини даних та ефективності охолодження. Проведення тестів у режимі повного навантаження дозволяє визначити наявність артефактів, помилок рендерингу або спотворень у кольоровій палітрі, що свідчить про потенційні порушення в роботі пристрою.

У свою чергу, підсистема оперативної пам'яті вимагає високоточного тестування на предмет цілісності переданих і збережених даних. Для цього застосовуються алгоритми багаторазового запису, зчитування та верифікації шаблонів різного типу, що дозволяє виявити логічні помилки, порушення в адресному просторі, деградацію елементів зберігання інформації або проблеми синхронізації з контролером пам'яті. У випадку систем з високою частотою та низькими таймінгами зростає ризик збоїв, які проявляються лише в умовах інтенсивного

багатозадачного навантаження, що вимагає застосування стрес-тестів із тривалим часом виконання.

Жорсткі диски (HDD) характеризуються специфічними типами відмов, які можуть бути пов'язані як з фізичним зношенням носія, так і з логічними помилками файлової системи. Технології тестування в цьому випадку орієнтовані на оцінювання швидкості доступу до даних, часу пошуку, стабільності читання/запису у випадкових і послідовних режимах, а також моніторинг показників SMART, що дозволяють прогнозувати ймовірність відмови. Особливої уваги потребують твердотільні накопичувачі з інтерфейсами NVMe, які, попри високу продуктивність, є вразливими до перегріву й нестабільності при інтенсивному використанні.

Підсистеми живлення, включно з блоком живлення та VRM-модулями материнської плати, також підлягають тестуванню, оскільки нестабільність напруг або їхнє коливання в межах критичних величин можуть спричинити нестабільність роботи всього комп'ютера. Методи тестування в цьому контексті базуються на вимірюванні напруг, частот комутації, температури силових компонентів, а також симуляції навантаження за допомогою штучних тестових стендів або бенчмарків.

Під оптимізацією апаратних компонентів персонального комп'ютера розуміється сукупність технічних і програмних заходів, спрямованих на покращення їхньої продуктивності, енергоефективності або стабільності з урахуванням специфіки конкретного сценарію експлуатації. Одним із найпоширеніших прикладів оптимізації є розгін, який передбачає збільшення тактової частоти роботи процесора, графічного ядра або оперативної пам'яті понад номінальні значення. Хоча такий підхід дозволяє досягти приросту продуктивності, він супроводжується зростанням тепловиділення та навантаження на систему живлення, що вимагає перегляду параметрів охолодження, напруг живлення та навіть оновлення прошивки BIOS.

Крім традиційного ручного розгону, в сучасних комп'ютерних системах використовуються алгоритми динамічного регулювання частот і напруг на основі прогнозованого навантаження. Такі технології, як Intel Turbo Boost або AMD Precision Boost, дозволяють тимчасово збільшити продуктивність одного або кількох ядер процесора в межах теплового пакета, заданого виробником. Аналогічні механізми діють у графічних процесорах, де можливе короткочасне підвищення частоти за умови відповідного теплового резерву. Оптимізація в цьому разі реалізується через інтелектуальні системи керування живленням, які постійно моніторять температурні та енергетичні показники й адаптують параметри роботи.

Не менш важливим аспектом оптимізації є узгодженість роботи компонентів системи між собою. Наприклад, використання швидкісної пам'яті в парі з процесором, архітектура якого не підтримує відповідні таймінги або частоти, може призвести до нестабільності, тоді як правильна синхронізація всіх параметрів дозволяє досягти максимального потенціалу системи. Аналогічна ситуація спостерігається у випадку роботи відеокарти з дисплеєм, що підтримує технологію змінної частоти оновлення, такою як G-Sync або FreeSync – коректне налаштування цих параметрів дає змогу уникнути ефекту «розриву кадру» та зменшити затримку виведення зображення.

Висновки. У сучасних умовах стрімкого розвитку комп'ютерних технологій ефективне тестування та оптимізація апаратних компонентів персональних комп'ютерів є критично важливими для забезпечення їхньої надійної та продуктивної роботи. Високий рівень складності апаратної архітектури вимагає застосування точних, системних підходів до діагностики, використання спеціалізованих інструментів аналізу продуктивності, а також інтеграції інтелектуальних засобів управління ресурсами.

Список літератури

1. Amit Bhanushali. "Evolving Trends in Quality Assurance Testing: A Comprehensive Review of Methodologies and Tools". International Journal of Advances in Engineering Research; Vol 26, 2023. Issue 4; 19-38.

2. Shtanenko, S., Samokhvalov, Y., Iohov , O.. "Microprocessor systems based on programmable logic devices as an object of diagnostics". *Advanced Information Systems*, *6*(1), 2022, 81–87.

Стрельчук В.А.[(1)], Гладій О.І. [(1)]
1) студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Мельник Г.М., кафедра КІ ЗУНУ

МОДЕЛЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ ПРИСТРОЇВ У WI-FI МЕРЕЖАХ

Вступ. Сучасні бездротові мережі Wi-Fi стали невіддільною складовою цифрової інфраструктури підприємств, установ, освітніх закладів та житлових будинків. Збільшення кількості підключених пристроїв, поява нових класів IoT-рішень, а також зростаючі вимоги до безпеки та стабільності мережевих з'єднань зумовлюють потребу у ефективному моніторингу активності в Wi-Fi-середовищі [1]. Моніторинг дозволяє виявляти сторонні підключення, аномальну поведінку пристроїв, перебої у зв'язку, а також допомагає в оптимізації використання ресурсів мережі. Особливого значення набувають алгоритмічні методи обробки трафіку та подій, які дозволяють здійснювати адаптивний аналіз у реальному часі.

Незважаючи на наявність комерційних рішень, залишається актуальним завдання розробки відкритих та масштабованих алгоритмів, що здатні інтегруватися у наявну мережеву інфраструктуру, зокрема з використанням обладнання типу MikroTik, UniFi або OpenWRT.

Постановка задачі. Метою дослідження є аналіз та формалізація алгоритмів моніторингу пристроїв у Wi-Fi мережах з урахуванням особливостей трафіку, властивостей клієнтських пристроїв та структури мережі. Необхідно розробити модель, яка дозволить: виявляти невідомі або потенційно небезпечні пристрої, будувати профілі типового використання Wi-Fi ресурсів.

Об'єкт дослідження: безпровідна комп'ютерна мережа стандарту Wi-Fi з множиною підключених пристроїв різних типів. Предмет дослідження**:** алгоритми збору, обробки та інтерпретації даних моніторингу пристроїв у Wi-Fi мережах.

Основний матеріал. Сформулюємо вимоги до розробки програмного засобу, призначеного для моніторингу активності пристроїв у Wi-Fi мережі на базі маршрутизаторів MikroTik з операційною системою RouterOS. Така система має реалізовувати наступні функції. Виявлення активних пристроїв у реальному часі з використанням вбудованих можливостей RouterOS (таблиця DHCP, ARP-таблиця, wireless registration table, hotspot active users тощо). Фіксація моментів підключення та відключення, включаючи ведення історії сеансів доступу з зазначенням MAC- та IP-адрес, часу та тривалості сесії. Оцінка інтенсивності взаємодії пристроїв із мережею шляхом обробки статистики трафіку: кількість вхідних/вихідних пакетів, обсяг переданої інформації, частота запитів до мережі, типи використовуваних протоколів (HTTP, DNS, TCP, UDP). Збір та агрегація даних із різних джерел RouterOS, включно з інструментами torch, ip accounting, interface monitor-traffic тощо, з подальшою передачею інформації у зовнішню систему обробки або візуалізації (через API, SNMP, syslog або інші інтерфейси).

Інтерфейсні та інтеграційні вимоги до програми передбачають реалізацію веб-інтерфейсу для доступу до результатів моніторингу, а також можливість інтеграції із зовнішніми системами безпеки, журналювання або з аналітичними платформами.

Об’єктну модель програмного засобу моніторингу активності пристроїв можна реалізувати у середовищі Python із використанням бібліотеки routeros-api [2] або librouteros [3]. Центральним компонентом є клас RouterClient, який інкапсулює функціонал підключення до RouterOS через протокол API, зберігаючи конфігурацію підключення (IP-адреса, порт, облікові дані користувача, параметри TLS або SSH-тунелювання для безпечного з’єднання). Клас містить методи get_active_devices(), get_traffic_stats(), get_connection_times(), які реалізують збір даних із таблиць /ip/hotspot/active, /ip/dhcp-server/lease, /interface/monitor-traffic, [/ip/firewall/connection], з можливістю фільтрації результатів та агрегації інформації за часовими вікнами. Питання безпеки реалізується на рівні автентифікації з обмеженими правами доступу до API, використанням IP-фільтрації на стороні RouterOS та опціональним шифруванням з’єднання. Для захисту конфіденційної інформації передбачено шифрування облікових даних у локальному конфігураційному файлі та ведення журналу доступу з верифікацією спроб входу. Клас

DeviceActivity моделює окремий пристрій у мережі з властивостями MAC-адреси, IP-адреси, часу останнього з'єднання, обсягу переданого трафіку, частоти звернень тощо. Обробка аномалій реалізується за допомогою класу AnomalyDetector, який аналізує відхилення від типового шаблону активності та формує події. Для реалізації сповіщень використовується модуль Notifier, який підтримує надсилання повідомлень через електронну пошту (SMTP) або інтеграцію з SIEM-системами через syslog чи HTTP-запити. Система забезпечує масштабованість завдяки підтримці асинхронного виконання запитів (через asyncio) та можливості паралельного моніторингу кількох пристроїв. Об'єктна модель (рисунок 1) спроектована з урахуванням принципів розширюваності, що дозволяє додавати нові типи джерел даних RouterOS та алгоритми обробки без порушення основної логіки роботи системи.

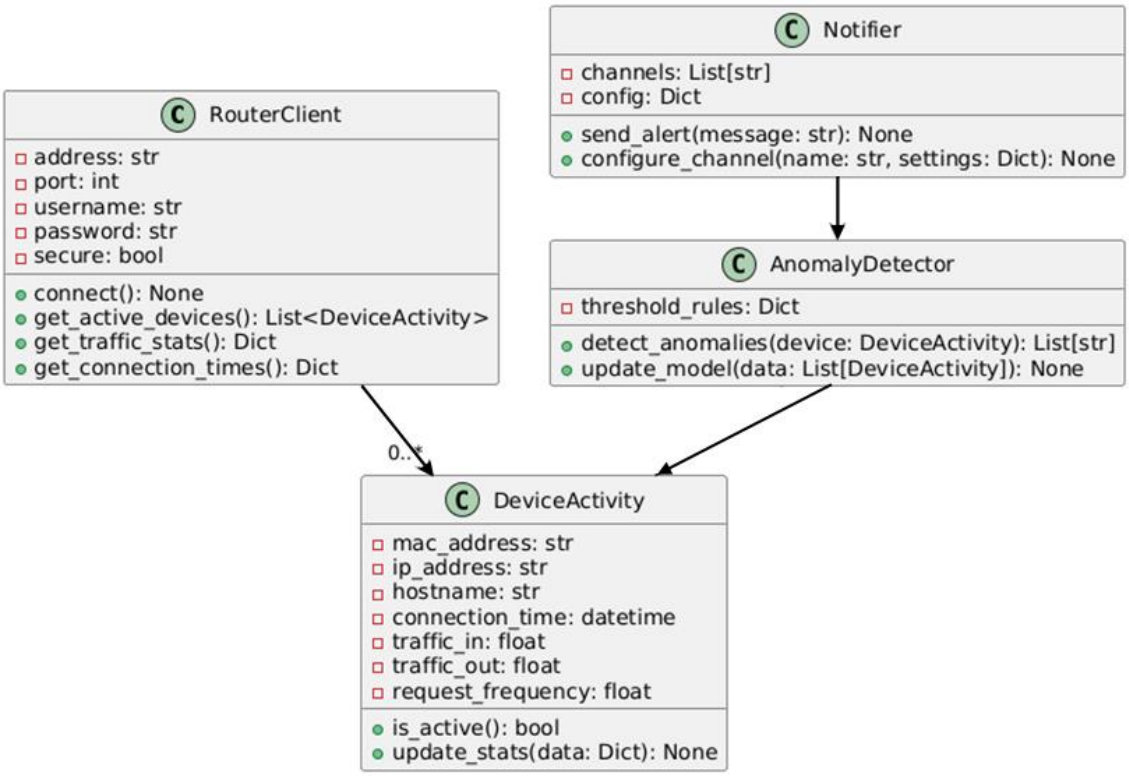


Рисунок 1 – Об'єктна модель

Опис діаграми на рисунку 1. Клас RouterClient забпезпечує інтерфейс до пристрою RouterOS. Клас DeviceActivity є моделлю активного пристрою з мережевою статистикою. Клас AnomalyDetector аналізує активності пристроїв і виявлення нетипової поведінки. Клас Notifier потрібен для надсилання сповіщень про події чи аномалії.

Висновки. Розроблена модель охоплює ключові компоненти для безпечного збору, обробки та аналізу даних про мережеву активність з урахуванням вимог до масштабованості, точності й інтеграції. Актуальність запропонованого підходу обумовлена зростанням кількості пристроїв у корпоративних і побутових мережах, необхідністю захисту периметру та мінімізації ризиків несанкціонованого доступу.

Список літератури

1. Jung Jin H., Rahimi Ghashghaei F., Elmrabit N. et al. Enhancing Sniffing Detection in IoT Home Wi-Fi Networks: An Ensemble Learning Approach With Network Monitoring System (NMS). IEEE Access. Vol. 12, 2024. P. 86840–86853. DOI:10.1109/ACCESS.2024.3416095.

2. socialwifi/RouterOS-api. Python API to RouterBoard devices produced by MikroTik. Social WiFi, 2025. URL: https://github.com/socialwifi/RouterOS-api (accessed 12/05/2025).

3. Kostka Ł. luqasz/librouteros. Python implementation of MikroTik RouterOS API. URL: https://github.com/luqasz/librouteros (accessed 12/05/2025).

Мельник Є.А.
студент групи КН-31сп
Національний університет «Львівська політехніка»

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

Вступ. З розвитком штучного інтелекту постає питання щодо його використання у різних сферах застосування. Зокрема, при застосуванні технологій з розпізнавання об'єктів шляхом аналізу комп'ютерної обробки візуальної інформації. Розпізнавання об'єктів на зображеннях — процес, за допомогою якого їх можна знайти і класифікувати. Застосування цього процесу швидко зростає у зв'язку з розвитком робототехніки, медичної діагностики, охорони, безпеки та інших галузей. З використанням штучного інтелекту з'являється можливість швидкої обробки вихідних даних та отримання високої точності результатів. Поряд з тим, для ефективного розпізнавання об'єктів потрібні великі масиви даних із анотованими зображеннями, що дає змогу ідентифікувати об'єкти на зображенні.

Постановка задачі. Об'єкт дослідження – процес використання згорткових нейронних мереж для аналізу візуальних зображень. Предмет дослідження – методи та засоби штучного інтелекту для розпізнавання автомобілів та номерних знаків на зображеннях. Стає важливим вибір моделі, яка б відповідала вимогам як за швидкістю, так і за точністю по виявленню об'єктів. Однією з таких найпопулярніших моделей є YOLO. Вперше представлена Джозефом Редмоном та ін. у 2015 році. YOLO зазнала кількох ітерацій та вдосконалень, ставши однією з найпопулярніших систем виявлення об'єктів [2].

Основний матеріал. На даний час існують різні методи і системи пошуку для розпізнавання об'єктів на зображеннях. Найпопулярнішими з них є методи з використанням згорткових нейронних мереж. Згорткові нейронні мережі (ЗНМ) — клас глибоких штучних нейронних мереж прямого поширення, який успішно застосовувався до аналізу візуальних зображень. ЗНМ використовують різновид багатошарових перцептронів, розроблений так, щоби вимагати використання мінімального обсягу попередньої обробки[1].

Розглянемо архітектуру моделі YOLO. Модель YOLO (рис. 1) використовує CNN-архітектуру, яка має декілька блоків під назвою Convolutional Layer (блок згорткового шару), в яких виконуються операції згортки та узагальнення. Кожна комірка в сітці передбачає певну кількість обмежувальних рамок. Поряд з кожною обмежувальною рамкою комірка передбачає ймовірність класу, яка вказує на ймовірність присутності певного об'єкта в рамці. Дана система розпізнавання має 24 згорткові шари, за якими йдуть 2 повністю зв'язані шари. Чергування згорткових шарів 1 x 1 зменшує простір ознак з попередніх шарів[4].

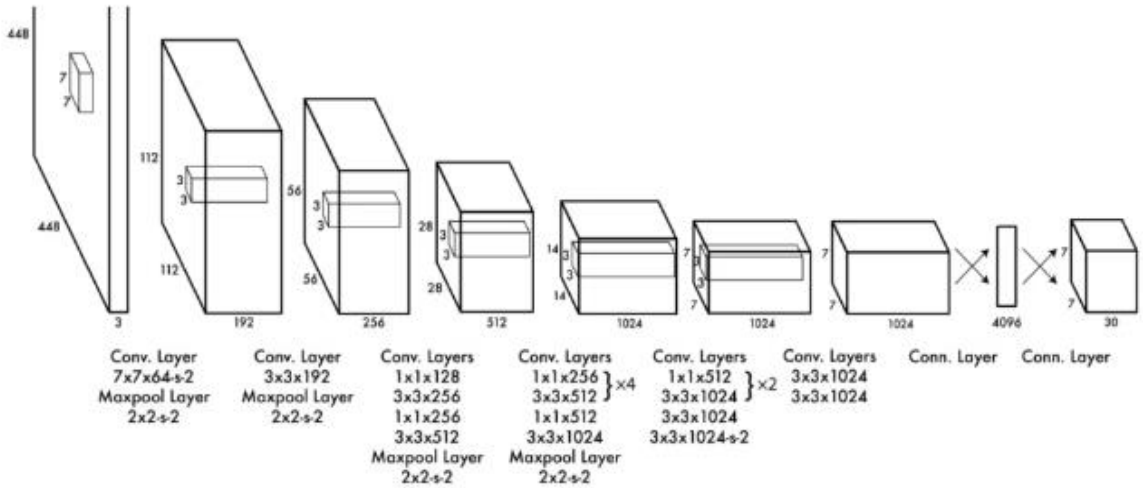


Рисунок 1. Архітектура YOLO.

Принцип роботи: зображення ділиться на сітку (наприклад, 7×7)[5]. Кожна клітинка прогнозує координати рамки та ймовірності класів, причому всередині однієї клітинки може бути не більше двох об’єктів одного класу.

До основних переваг YOLO відносять:

- Швидкість . Вона може обробляти зображення зі швидкістю 45 кадрів на секунду (FPS). З графіка нижче видно, що YOLO значно перевершує інші моделі виявлення об'єктів, маючи 91 FPS(рис.2) [3].

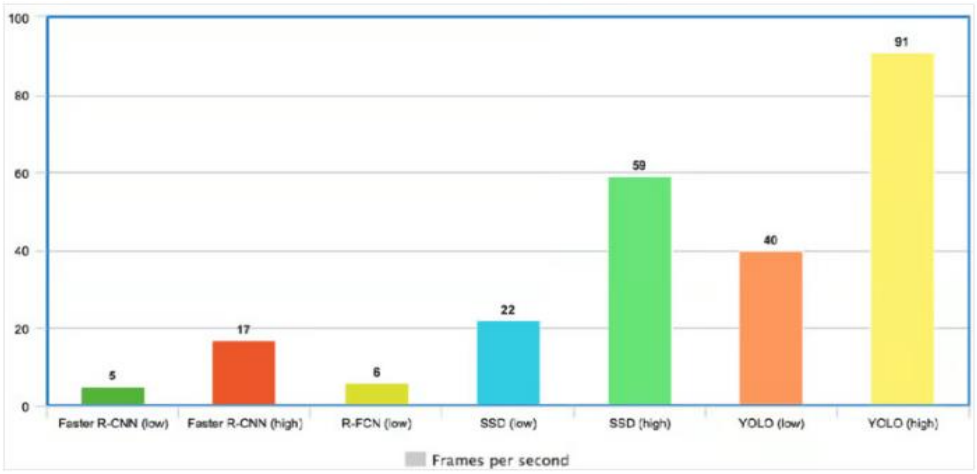


Рисунок 2 - Швидкість YOLO порівняно з іншими сучасними моделями виявлення об'єктів

- Висока точність виявлення. Постійні вдосконалення архітектури та методів навчання, YOLO демонструє високі показники mAP (середній показник точності) на стандартних наборах даних, часто перевершуючи інші одностадійні моделі та конкуруючи з двостадійними моделями. Модель YOLO під час проведення тестування, перевершує інші сучасні моделі такі, як Faster R-CNN, R-FCN та SSD, а також модель YOLO містить досить низьку кількість помилок при розпізнаванні об’єктів[3].
- Відкритий код. Для спільноти розробників та дослідників, вихідний код моделі YOLO є у відкритому доступі[3]. Завдяки цьому, модель постійно вдосконалюється і отримує значну кількість покращень за короткий проміжок часу.

Еволюційний шлях розвитку YOLO на сьогоднішній день представлено новітньою версією серії YOLOv12 (2025рік). Порівняльну характеристику усіх версій наведено у таблиці 1.

Таблиця 1 Порівняльна характеристика

Версія (рік)	**Архітектура (Backbone)**	**Точність mAP**	**Швидкодія (FPS)**	**Код (реалізація)**	**Основні сфери використання**
YOLOv1 (2016)	24 conv + 2 FC (GoogLeNet) [4]	~63.4% (PASCAL VOC2007	~45 FPS (Titan X)	Darknet (C)	Детекція об’єктів (відеоспостереження, робототехніка)
YOLOv2 (2017)	Darknet-19 (19 conv+5 pool) [4]	78.6% (PASCAL VOC2007)	67 FPS (416×416)	Darknet (C)	Детекція об’єктів у реальному часі, відеоспостереження
YOLOv3 (2018)	Darknet-53 [4]	57.9% (COCO test-dev)	~30–45 FPS	Darknet (C)	Різноманітні прикладні задачі CV (відеодетекція тощо)
YOLOv4 (2020)	CSPDarknet53 [5]	43.5% AP (COCO)	~65 FPS	Darknet (C)	Інтеграція BoF/BoS методів, реальне-часу детекція

Версія (рік)	Архітектура (Backbone)	Точність mAP	Швидкодія (FPS)	Код (реалізація)	Основні сфери використання
YOLOv5 (2020)	CSPDarknet53 (з C3/SPPF) [6]	~50–52% (COCO, найкращі версії)	до 140 FPS	PyTorch (Ultralytics)	Широкий спектр CV-задач (детекція, адаптивність до різних задач)
YOLOv6 (2022)	RepCSP (власний) [7]	37.5–52.8% (COCO)	116–1187 FPS	PyTorch (Ultralytics)	Мобільні додатки, вбудовані системи (оптимізований для швидкості)
YOLOv7 (2022)	E-ELAN (ELAN із вдосконаленнями) [4]	51.4–56.8% (COCO)	5–165 FPS	PyTorch (Ultralytics)	Максимальна ефективність: детекція, сегментація, керування дронами тощо
YOLOv8 (2023)	Модифікований CSPDarknet53 + C2 [4]	~51% (COCO, найкраща версія)	до 280 FPS	PyTorch (Ultralytics)	Універсальний (детекція, сегментація, поза), широкий вжиток
YOLOv9 (2024)	GELAN + C3Ghost (PGI, etc.) [8]	>50% (COCO, покращення над YOLOv8)	дуже висока (компроміс між FPS та mAP)	PyTorch (Ultralytics)	Інтернет речей, вбудовані системи (висока ефективність)
YOLOv10 (2024)	Сучасний дизайн (NMS-free, оптимізовані блоки)[7]	~ (SOTA на рівні попередників)	значно швидший (YOLOv10-S 1.8× швидше за RT-DETR)	PyTorch (Ultralytics)	Енд-ту-енд детекція, мінімізовані затримки (для складних систем реального часу)
YOLOv11 (2024)	Покращений бекбон та шийка (CSP+ELAN) [7]	Вищий (за рахунок нових оптимізацій)	+25% швидкості порівняно з v10	PyTorch (Ultralytics)	Багатозадачність (детекція, сегментація, поза, обернені бокси), впровадження на edge
YOLOv12 (2025)	Attention-centric (Area Attention + R-ELAN)l[7]	40.6% (COCO, YOLOv12-N)	~610 FPS	PyTorch (Ultralytics)	Універсальні CV-застосування: детекція, сегментація з урахуванням уваги

Отож, архітектура YOLO за кілька років пройшла шлях від базової мережі з 24 згортками до складних модулів з CSP-блоками. В результаті YOLO залишається одним із найпопулярніших підходів для вирішення задач, де необхідна висока швидкість і точність обробки зображень, що знайшло своє застосування в різних галузях науки та техніки.

Для прикладу проведемо експеримент з практичного застосування моделі YOLO при розпізнаванні на зображеннях автомобілів та їх номерних знаків.

На початку формуємо датасет, шляхом збирання достатньої кількості зображень, на яких присутні автомобілі з номерними знаками. Далі всі об'єкти на зображеннях анотуються - накреслюються обмежувальні рамки для кожного класу ("car", "license_plate/licence") за допомогою зручних онлайн-сервісів, наприклад, як Roboflow або LabelImg.txt

Розділяємо наш датасет на дві окремі папки train і val (тренувальну і валідаційну вибірки). Створюємо YAML-файл конфігурації датасету (наприклад, dataset.yaml), де вказано шляхи до папок train/val, кількість класів та список назв класів. Структура конфігураційного файлу буде виглядати наступним чином (див. рис.3.)

```
dataset.yaml
1
2 train: /content/drive/MyDrive/archive_unzip/archive/images/train
3 val: /content/drive/MyDrive/archive_unzip/archive/images/val
4 nc: 2
5 names: ['car', 'licence']
6
```

Рисунок 3 Структура YAML-файлу

Обираємо модель YOLO (наприклад, YOLO11), яку будемо тренувати. Для цього завантажуємо pre-trained (базову) модель, що вже навчена на великому датасеті (COCO). Починаємо процес донавчання моделі на нашому підготовленому датасеті з двома класами: “car” і “license_plate” (або “licence”). Модель вчиться розпізнавати ці об’єкти на зображеннях.

Після завершення навчання моделі, проводимо її тестування на зображеннях. Отриманий результат відображено на рис. 4.

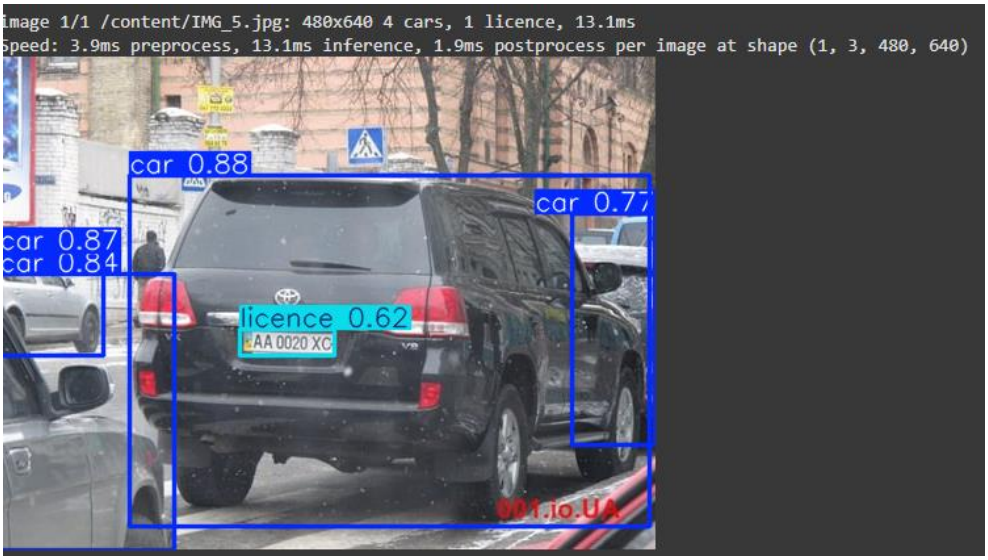


Рисунок 4 Результат застосування моделі YOLOv11 при розпізнаванні на зображеннях автомобілів та їх номерних знаків

Висновки. Отже, натренована модель YOLOv11 на підготовленому датасеті, продемонструвала високі результати точності, що свідчить про її надійність та ефективність для вирішення практичних задач з розпізнавання об’єктів. Запропонований підхід може бути використаний у різних сферах: від контролю дорожнього руху до систем безпеки та паркування.

Список літературних джерел

1. Wikipedia. Згорткова нейронна мережа - uk.wikipedia.org https://uk.wikipedia.org/wiki/Згорткова_нейронна_мережа.
2. Wikipedia. You_Only_Look_Once - en.wikipedia.org https://en.wikipedia.org/wiki/You_Only_Look_Once.
3. Sanchez S. A., Romero H. J., Morales A. D. A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. IOP Conf. Ser.: Mater. Sci. Eng., 2020, vol. 844, 012024. DOI: 10.1088/1757-899X/844/1/012024
4. Vijayakumar A., Vairavasundaram S. YOLO-based Object Detection Models: A Review and its Applications. Multimedia Tools and Applications, 2024. DOI: 10.1007/s11042-024-18872-y
5. Terven J., Córdova-Esparza D.-M., Romero-González J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extr., 2023, 5(4), 1680–1716. DOI: 10.3390/make5040083
6. Handalage U., Kuganandamurthy L. Real-Time Object Detection Using YOLO: A Review. – May 2021.
7. Brief Review: YOLOv5 for Object Detection - https://shtsang.medium.com/brief-review-yolov5-for-object-detection-84cc6c6a0e3a.
8. Ultralytics YOLO - https://docs.ultralytics.com/models/yolov6/#key-features.
9. Yaseen M. What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector // arXiv:2409.07813v1 [cs.CV], 12 Sep 2024. – Available at: https://arxiv.org/abs/2409.07813

Шнайдрук Д.О.
студент 4 курс ФКІТ ЗУНУ
Науковий керівник к.т.н. Паздрій І.Р. кафедра КІ ЗУНУ

ПРИСТРІЙ КЕРУВАННЯ ТА ЕЛЕКТРОЖИВЛЕННЯ СВІТЛОДІОДНИХ ДЖЕРЕЛ СВІТЛА

Вступ. Світлодіодні (LED) освітлювальні системи демонструють значні переваги над традиційними джерелами світла, зокрема, вищу енергоефективність, довший термін служби та покращені можливості керування. Ключовим елементом таких систем є LED-драйвер, що забезпечує стабільне живлення та керування світлодіодами. Актуальність роботи полягає у розробці та дослідженні LED-драйвера з мікроконтролерним керуванням, що дозволяє не лише точно регулювати яскравість світлового потоку методом широтно-імпульсної модуляції (ШІМ), але й створює платформу для реалізації інтелектуальних функцій освітлення та підвищення загальної енергоефективності системи.

Постановка задачі. Завданням роботи є комплексне дослідження, розробка схемотехнічного рішення, створення та експериментальна верифікація прототипу LED-драйвера з мікроконтролерним керуванням, здатного забезпечити стабілізоване живлення світлодіодного навантаження та плавне регулювання яскравості в широкому діапазоні.

Об’єкт дослідження - процес керування та електроживлення світлодіодних джерел світла.

Предмет дослідження - LED-драйвер з мікроконтролерним керуванням.

Основний матеріал. Мікропроцесорні струмові регулятори широко застосовуються для керування рівнем яскравості освітлення, створення світлових ефектів та регулювання інтенсивності світлового потоку. Зазвичай такі регулятори характеризуються низьким керувальним струмом. У випадках, коли необхідно керувати світлодіодами з підвищеним струмом, до системи додається додатковий силовий елемент — спеціалізований блок живлення для світлодіодів.

Більшість LED-драйверів оснащені дозвільними входами, які можуть бути як аналоговими, так і цифровими. Це дозволяє регулювати рівень живлення світлодіодів, передаючи сигнал від модулятора ШІМ безпосередньо з виходу мікроконтролера. При цьому важливо враховувати максимально допустиму частоту ШІМ-сигналу, яка може бути обмеженням для коректної роботи дозволу живлення.

Застосування мікросхем із розширеним діапазоном вхідної напруги суттєво полегшує процес проектування, оскільки дозволяє зменшити або повністю уникнути використання додаткових захисних компонентів.

Пропонується LED-драйвер із мікроконтролерним управлінням, який забезпечує стабільне живлення світлодіодного навантаження та дозволяє плавно регулювати яскравість у широкому діапазоні. Основні функціональні блоки реалізовані як програмними засобами, так і апаратно за допомогою мікроконтролера STM32F103C8T6 [4]. LED-драйвер поєднує силову частину на базі Buck-конвертера та цифрову систему керування на мікроконтролері. Для генерації широтно-імпульсної модуляції використовується вбудований апаратний 8-розрядний таймер-лічильник. Аналого-цифрове перетворення виконується за допомогою внутрішнього АЦП мікроконтролера, який працює за принципом послідовного наближення та налаштований на 8-бітний режим.

В якості основи силової частини було обрано топологію неізольованого понижуючого перетворювача, що працює в режимі стабілізації струму. Керування здійснюється мікроконтролером STM32F103C8T6 на базі ядра ARM Cortex-M3, який генерує ШІМ-сигнал для керування силовим MOSFET-транзистором IRFZ44N.

Зворотний зв'язок по струму реалізований за допомогою низькоомного шунтового резистора та операційного підсилювача. Регулювання яскравості здійснюється користувачем за допомогою інкрементального енкодера.

Програмне забезпечення розроблено в середовищі STM32CubeIDE мовою C. Моделювання перехідних процесів та статичних характеристик силової частини проводилося в LTSpice.

Експериментальні вимірювання проводилися з використанням цифрового осцилографа Rigol DS1102E, мультиметра Fluke 179, електронного навантаження та лабораторного джерела живлення. Результати показали очікувану стабільність роботи перетворювача та можливість ефективного керування вихідним струмом за допомогою ШІМ. Результати моделювання показали очікувану стабільність роботи перетворювача та можливість ефективного керування вихідним струмом за допомогою ШІМ. Ключовим графіком, отриманим в результаті моделювання в середовищі LTSpice, є часова діаграма, що відображає перехідний процес встановлення та стабілізації вихідного струму через світлодіодне навантаження при подачі живлення та зміні скважності керуючого ШІМ-сигналу.

Початковий етап наростання струму в дроселі та через світлодіоди триває до заданого рівня 700 мА. Стабілізація вихідного струму на цьому рівні з незначними пульсаціями, амплітуда яких відповідає розрахунковим значенням ±25 мА. При зменшенні скважності ШІМ з 70% до 30%, спостерігається плавне зниження середнього значення вихідного струму до відповідного меншого рівня до 300 мА, що демонструє ефективність реалізованого димування. Струм дроселя, який має характерну пилкоподібну форму, підтверджує роботу перетворювача в неперервному режимі дроселя або дискретному, залежно від розрахунку. Схема підтверджує коректність обраних параметрів силових елементів (індуктивності дроселя, ємності вихідного конденсатора) та працездатність алгоритму керування. Написаний програмний код для мікроконтролера забезпечує генерацію ШІМ-сигналу з роздільною здатністю 8-10 біт та частотою 10-20 кГц, що мінімізує видиме мерехтіння та акустичний шум.

Висновок. Результати експериментів підтвердили очікувану працездатність схеми. Розробленмй та експериментально досліджений прототип LED-драйвера з мікроконтролерним керуванням підтвердив свою працездатність та ефективність. Використання мікроконтролера STM32 та ШІМ-методу дозволяє забезпечити точне і плавне регулювання яскравості світлодіодів у широкому діапазоні при високому ККД.

Отримані експериментальні дані добре узгоджуються з відомими результатами моделювань, що підтверджує адекватність використаних моделей. Порівняльний аналіз з існуючими аналогами та теоретичними розрахунками показав високу ступінь відповідності отриманих експериментальних даних, що підтверджує адекватність обраної схемотехніки та методики розрахунку. Розроблений драйвер продемонстрував переваги у гнучкості налаштувань та точності регулювання яскравості завдяки мікроконтролерному керуванню, порівняно з базовими аналоговими або дискретними рішеннями, а також досягнув конкурентоспроможних показників ККД у своєму класі потужності.

Розроблений пристрій має потенціал для подальшого вдосконалення, зокрема, шляхом додавання функцій захисту від перенапруги та обриву навантаження, реалізації бездротових інтерфейсів керування Bluetooth, Wi-Fi та інтеграції з датчиками для автоматичного регулювання освітленості.

Список літератури.

1. Журахівський Ю.П., Поліщук О.С. *Силова електроніка: навчальний посібник.* Київ: НТУУ "КПІ", 2018. – 240 с.

2. Basso, C. P. *Switch-Mode Power Supplies: SPICE Simulations and Practical Designs.* McGraw-Hill Professional, 2018. – 896 p.

3. Тревор Мартін. *Микроконтроллеры STM32 для начинающих. Полное руководство.* Київ: Діалектика, 2021. – 304 с. (Приклад книги по STM32)

4. STMicroelectronics. *STM32F103C8T6 Performance line, ARM-based 32-bit MCU with 64 Kbytes Flash. Datasheet (DocID013587 Rev 20)*. [Електронний ресурс]. Доступно: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf

Антонюк В.О.[1], Мельник Г.М. [2]
(1) студент 4 курсу ФКІТ ЗУНУ, (2) доцент кафедри КІ ЗУНУ
Науковий керівник к.т.н., доцент Мельник Г.М., кафедра КІ ЗУНУ

АРХІТЕКТУРА ПРОГРАМНО-АПАРАТНИХ СИСТЕМ КЕРУВАННЯ РЕЗЕРВНИМ ЖИВЛЕННЯМ БУДИНКУ

Вступ. Нестабільність електропостачання, зростання вартості енергії та впровадження відновлюваних джерел живлення потребують створення ефективних систем управління резервними джерелами енергії. Актуальним є аналіз архітектури програмно-апаратних систем, що інтегрують джерела резервного живлення, контролери, датчики та програмне забезпечення. Такі системи дозволяють забезпечити надійність, оптимізувати використання енергії та підвищити рівень автоматизації в управлінні електропостачанням.

Постановка задачі. Метою роботи є аналіз архітектури програмно-апаратних систем керування резервним живленням для "розумних" будинків, визначення основних компонентів, принципів побудови та типових підходів, що використовуються у сучасних рішеннях. Завдання полягає у виявленні загальної структури системи, взаємодії програмних і апаратних елементів, а також характеристик, що забезпечують надійність, масштабованість та ефективність управління джерелами резервного живлення. Об'єктом дослідження є архітектура програмно-апаратних систем керування резервним живленням. Предметом дослідження є алгоритми обчислення та прогнозування параметрів роботи системи резервного живлення.

Основний матеріал. Архітектура програмно-апаратної системи керування резервним живленням "розумного" будинку має модульну структуру та включає такі ключові компоненти. Джерела резервного живлення забезпечують електропостачання у разі відключення основної мережі (акумуляторні батареї, генератори, системи на базі відновлюваних джерел енергії). Контролери виконують функції збору даних, моніторингу стану системи та управління процесами перемикання між джерелами живлення. Сенсори для вимірювання електричних параметрів (струм, напруга, рівень заряду) та умов навколишнього середовища. Комунікаційні модулі для обміну даними між компонентами системи, часто з використанням протоколів MQTT, Modbus, ZigBee або Wi-Fi. Програмне забезпечення реалізує алгоритми управління, обробку даних та надає користувацький інтерфейс для моніторингу і керування системою (наприклад, мобільні додатки, веб-панелі).

Архітектура програмно-апаратної системи керування резервним живленням "розумного" будинку ґрунтується на кількох ключових принципах (рисунок 1). Насамперед важливою є модульність, що дозволяє легко адаптувати систему до різних масштабів та сценаріїв використання, а також спрощує обслуговування та модернізацію компонентів. Іншим важливим принципом є масштабованість, завдяки якій можливо розширювати систему під потреби користувача, додаючи нові пристрої, датчики або джерела живлення без необхідності значної перебудови архітектури. Система також передбачає інтеграцію програмної та апаратної частин для забезпечення ефективного обміну даними і гнучкості управління. Крім того, особливу увагу приділено забезпеченню безпеки даних та доступу, що є критично важливим аспектом у середовищі мережевих і підключених до Інтернету систем.

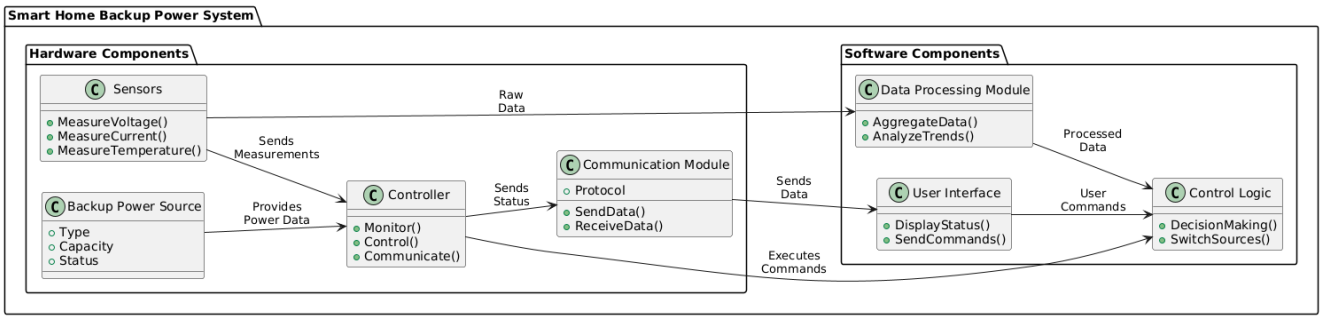


Рисунок 1 - Архітектура програмно-апаратної системи

UML-діаграма включає основні апаратні компоненти (джерела живлення, контролери, сенсори, комунікаційні модулі) та програмні частини (логіка керування, обробка даних, інтерфейси користувача).

До складу апаратних компонентів належать джерела резервного живлення, такі як акумуляторні батареї, генератори або системи на базі сонячних панелей. Контролери виконують функції управління системою, здійснюючи моніторинг параметрів та приймаючи рішення про перемикання джерел живлення. Сенсори забезпечують збір даних про електричні параметри, включаючи напругу, струм, температуру. Для обміну даними між компонентами використовуються комунікаційні модулі, що підтримують протоколи MQTT, Modbus, ZigBee або Wi-Fi. Програмні компоненти системи реалізують логіку управління резервним живленням, алгоритми перемикання між джерелами енергії, а також забезпечують обробку даних, їх збереження та візуалізацію. Інтерфейси користувача, як правило, представлені у вигляді мобільних додатків або веб-панелей. Інтерфейси дозволяють контролювати стан системи в реальному часі, отримувати сповіщення про аварійні ситуації та задавати параметри роботи системи.

Сучасний ринок програмно-апаратних систем керування резервним живленням "розумних" будинків представлений широким спектром рішень від провідних світових виробників. Серед найвідоміших вендорів варто відзначити Tesla, яка пропонує систему Powerwall – інтегроване рішення для домашнього зберігання енергії з можливістю дистанційного моніторингу та управління через мобільний додаток. Victron Energy спеціалізується на модульних системах з резервними джерелами живлення, контролерами та інверторами, що підтримують гнучке налаштування під потреби користувача. EcoFlow розробляє мобільні електростанції та системи для резервного живлення, орієнтовані на автономне використання у побутових умовах.

Також на ринку представлені рішення від таких компаній, як Schneider Electric, Huawei, SMA Solar та інших, що пропонують комплексні рішення для резервного живлення з інтеграцією у системи розумного будинку. Багато рішень підтримують хмарні сервіси для зберігання даних, аналітики та віддаленого доступу, що робить їх зручними у використанні та масштабуванні.

Однією з важливих функцій систем керування резервним живленням є обчислення та прогнозування параметрів роботи системи з урахуванням поточних і прогнозованих навантажень, стану джерел живлення та режимів споживання електроенергії. Такі системи повинні оцінювати рівень заряду акумуляторів, потужність споживання у різні періоди часу, залишковий ресурс джерел живлення та здатність системи підтримувати живлення в автономному режимі протягом певного інтервалу часу.

Для виконання цих завдань використовуються алгоритми моделювання навантаження за допомогою часових рядів, прогнозування споживання електроенергії на основі історичних даних, а також алгоритми визначення залишкового часу автономної роботи. Деякі системи застосовують методи машинного навчання для побудови моделей споживання енергії, виявлення аномальних режимів роботи або оптимізації роботи джерел резервного живлення з урахуванням змін умов навколишнього середовища.

Висновки. Функція прогнозування є критично важливою для забезпечення ефективного управління енергією: вона дозволяє користувачам отримувати попередження про можливе виснаження джерел живлення, оптимізувати розподіл навантаження та забезпечувати належну готовність системи до аварійних ситуацій.

Список літератури

1. Dinmohammadi F., Han Y., Shafiee M. Predicting Energy Consumption in Residential Buildings Using Advanced Machine Learning Algorithms. Energies. Vol. 16, № 9. P. 3748.

2. Gaikwad N., Lamichhane S., Dubey A. Model Predictive Control based Energy Management System for Home Energy Resiliency. arXiv, 2024.

3. Watil A. Smart home power management algorithm using real-time model predictive control for a stand-alone PV system with battery energy storage. e-Prime - Advances in Electrical Engineering, Electronics and Energy. Вип. 10, 12.2024. P. 100789.

Волох В.В. (1), Барановський В.О. (1)
(1) студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Дубчак Л.О., кафедра КІ ЗУНУ

ІЄРАРХІЧНА МОДЕЛЬ СИСТЕМИ МОНІТОРИНГУ В СИСТЕМІ THINGSBOARD

Вступ. Актуальність теми дослідження полягає в необхідності ефективного управління та моніторингу великої кількості пристроїв і даних, які генеруються в сучасних інфраструктурах розумних міст [1]. Інтенсивне впровадження технологій Інтернету речей [2] створює потребу в потужних та масштабованих програмних платформах, здатних забезпечити централізоване збирання, обробку, аналіз даних і своєчасне реагування на події. Важливою складовою таких систем є ієрархічна модель моніторингу, що дозволяє чітко організувати роботу з різнорідними пристроями, даними, правилами та тривогами, спрощуючи керування інфраструктурою розумного міста.

Постановка задачі. Метою роботи є дослідження та аналіз особливостей застосування ієрархічної моделі моніторингу системи Інтернету речей на базі платформи ThingsBoard, а також визначення основних етапів розгортання сервера моніторингу для розумного міста.

Для досягнення поставленої мети визначено такі основні завдання:

– опис основних елементів ієрархічної моделі (пристрої, профілі пристроїв, атрибути, ланцюги правил, тривоги, клієнти);

– розробка чіткої послідовності етапів створення сервера моніторингу в контексті розумного міста.

Об'єктом дослідження є програмна платформа ThingsBoard як базова система інтеграції, моніторингу й управління пристроями Інтернету речей в інфраструктурі розумного міста.

Предметом дослідження є ієрархічна модель моніторингу, включаючи структуру даних, управління пристроями, профілі, правила, сигнали тривог і організацію користувачів.

Основний матеріал. Ієрархічна модель організації системи моніторингу [3] базується на концепції взаємозв'язаних елементів, таких як пристрої (*devices*), профілі пристроїв (*device profiles*), сигнали тривог (*alarms*), ланцюги правил (*rule chains*) та користувачі (*customers*). Правильне налаштування цих елементів визначає ефективність роботи системи в контексті моніторингу інфраструктури розумного міста.

Пристрої (*Devices*) є базовими елементами системи, що збирають і передають телеметричні дані. Кожен пристрій може мати три типи атрибутів:

- Клієнтські (*client attributes*) – інформація, що надсилається безпосередньо з пристроїв, наприклад версія прошивки або стан батареї.
- Загальні (*shared attributes*) – інформація, що доступна як пристрою, так і серверу, наприклад, порогові значення для датчиків.
- Серверні (*server attributes*) – інформація, встановлена на сервері адміністратором, наприклад, місце розташування або дата технічного обслуговування пристрою.

Профіль пристрою (*Device Profile*) визначає шаблони конфігурації для пристроїв, включаючи тип пристрою, модель, налаштування для керування тривогами, правила обробки даних, формати телеметрії, а також конфігурації оновлення програмного забезпечення. Це дозволяє спростити масове додавання однотипних пристроїв у систему та керування ними.

Сигнали тривог (*Alarms*) забезпечують повідомлення адміністраторів та операторів про критичні події або ситуації, що вимагають втручання. Alarms генеруються на основі правил, встановлених в профілі пристрою або спеціальних ланцюгах правил. Вони дозволяють миттєво реагувати на позаштатні ситуації, наприклад, вихід показників за межі норми або втрату зв'язку з пристроєм. Ланцюги правил (*Rule Chains*) визначають послідовність обробки телеметрії та подій на сервері ThingsBoard. Це графічно налаштовуваний механізм, що дозволяє формувати логіку маршрутизації даних, обчислення додаткових метрик, генерації тривог, повідомлень та інтеграції з іншими системами. Правила забезпечують автоматизацію обробки даних відповідно до логіки бізнес-процесів розумного міста.

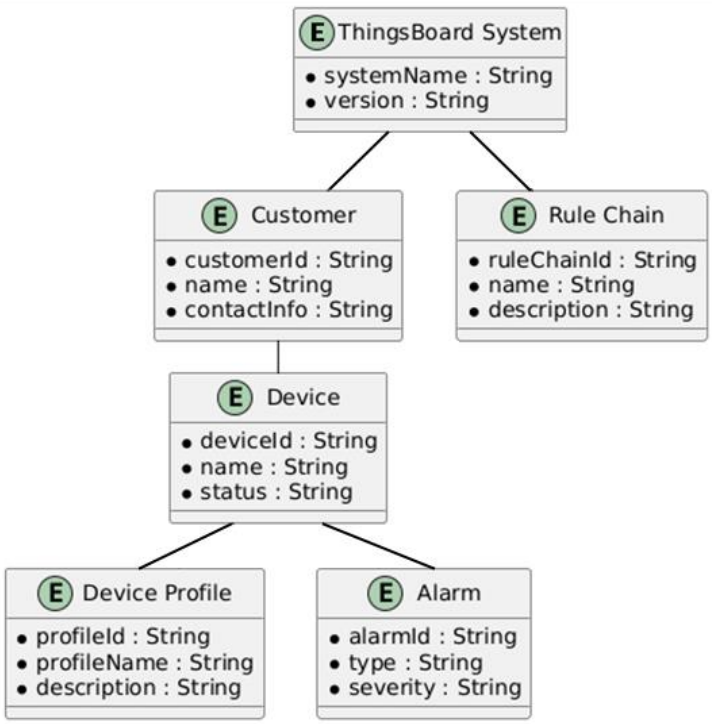


Рисунок 1 - Ієрархічна модель організації системи моніторингу

Користувачі (*Customers*) є кінцевими споживачами системи моніторингу. ThingsBoard дозволяє розподіляти пристрої та їх групи між різними організаціями чи відділами, надаючи користувачам окремий доступ до панелей керування, звітів і даних. Це дозволяє розмежовувати права доступу, забезпечуючи багаторівневу організацію роботи в умовах розумного міста.

Послідовність створення сервера моніторингу в контексті розумного міста на основі платформи ThingsBoard:

1. Розгортання сервера ThingsBoard локально чи в хмарі.
2. Створення профілів пристроїв, визначення типів сенсорів.
3. Реєстрація пристроїв та встановлення атрибутів.
4. Налаштування ланцюгів правил для автоматичної обробки, агрегації, генерації тривог та інтеграції із зовнішніми системами.
5. Створення системи сповіщень.
6. Реєстрація та налаштування прав доступу користувачів).
7. Формування аналітичних панелей для моніторингу і управління інфраструктурою міста.

Висновки. Проведено комплексне дослідження ієрархічної моделі моніторингу на базі платформи ThingsBoard та визначено основні етапи розгортання сервера для потреб інфраструктури розумного міста. Ієрархічна модель платформи ThingsBoard. забезпечує гнучку побудову системи для моніторингу різноманітних об’єктів і процесів у межах міської інфраструктури

Список літератури

1. Боголюбов В. М., Клименко М. О., Мокін В. Б. Моніторинг довкілля : підручник. ВНТУ. Вінниця, 2017. 232 с.

2 Okhrimchuk Ye. Monitoring the efficiency of existing industrial internet of things solutions. Vìsnik Sums′kogo deržavnogo unìversitetu. Vol. 2022, Issue 3. P. 97–105.

3. ThingsBoard Open-source IoT Platform Device management, data collection, processing and visualization for your IoT solution. URL: https://thingsboard.io

Костецький В.В.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

АЛГОРИТМИ ТА ЗАСОБИ ДЛЯ АНАЛІЗУ ЕМОЦІЙ У ВІДЕОПОТОЦІ

Вступ. Аналіз емоцій за відеопотоком є однією з актуальних задач в області комп'ютерного зору та обробки мультимедійних даних. Ця задача передбачає автоматичне виявлення і розпізнавання емоцій людини на основі зображень або послідовностей кадрів, отриманих із відео. Для навчання і тестування моделей, які здійснюють аналіз емоцій, використовуються спеціальні датасети, що містять анотовані приклади емоційних станів. Вибір відповідного датасету є важливим для побудови та перевірки моделей.

Постановка задачі. Метою роботи є дослідження алгоритмів і засобів аналізу емоцій у відеопотоці. Об'єктом дослідження є процес моделювання зміни емоцій на обличчі людини. Предметом дослідження є алгоритми і програмні засоби визначення алгоритми аналізу облич у відеопотоці.

Основний матеріал. Для аналізу емоцій на обличчях у відеопотоці використовуються різні публічно доступні датасети. AffectNet – один з найбільших датасетів для аналізу емоцій, містить понад 0,4 млн зображень з анотаціями семи базових емоцій. Хоча він орієнтований на статичні зображення, його можна адаптувати для роботи з відео. Real-world Affective Faces Database RAF-DB – містить близько 30 000 зображень облич із мітками емоцій. Як і AffectNet, використовується переважно для аналізу окремих кадрів. Спеціалізований датасет Emotion Recognition in the Wild EmotiW який включає відеокліпи з фільмів і реальних сцен, анотації для класифікації емоцій та задачі, пов'язані з мультимодальним аналізом. AFEW (Acted Facial Expressions in the Wild) – популярний датасет для розпізнавання емоцій у відеопотоці. Він містить короткі відеофрагменти зі сцен фільмів, де актори виражають різні емоції. CREMA-D (Crowd-Sourced Emotional Multimodal Actors Dataset) містить понад 7 тисяч аудіовізуальних кліпів, у яких актори озвучують однакові фрази з різними емоційними відтінками. DAiSEE (Dataset for Affective States in E-Environments) – фокусується на оцінці емоцій користувачів у навчальному середовищі, включаючи такі стани, як зацікавленість, задоволення, розчарування та нудьга.

Для завдань аналізу емоцій саме у відеопотоці варто розглянути датасети, що містять послідовності кадрів із часовою інформацією, такі як AFEW, EmotiW, CREMA-D або DAiSEE.

Для аналізу емоцій у відеопотоці використовуються різні формати зберігання даних. Зазвичай відео представлені у форматах mp4 або avi, оскільки вони забезпечують хорошу якість і сумісність із більшістю програмних засобів обробки відео. Анотації до відео, тобто інформація про емоції, які потрібно розпізнавати, зберігаються у вигляді таблиць або структурованих текстових файлів. Найчастіше використовують формати csv або json. У цих файлах вказують мітки емоцій для певних кадрів або ділянок відео, час або номер кадру, а також додаткову інформацію, наприклад, координати обличчя або рівень впевненості у визначеній емоції.

Для завдань, пов'язаних саме зі зміною емоцій у часі, важливо обрати такий формат даних, який дозволяє зберігати послідовність кадрів або часових міток разом із відповідними анотаціями. Це можуть бути файли json або csv, де кожному кадру або відрізку відео відповідає конкретна емоція, а також, за потреби, додаткова інформація про інтенсивність емоцій або координати обличчя. Окрім цього, для збереження проміжних результатів аналізу, наприклад, векторних ознак, що описують обличчя на кожному кадрі, часто використовують формати npy або h5, оскільки вони дозволяють ефективно зберігати великі масиви числових даних.

Формати npy та h5 (HDF5) використовуються у випадках, коли потрібно зберігати масиви числових даних, які виникають під час обробки відео. Наприклад, якщо ми аналізуємо емоції у відеопотоці, модель комп'ютерного зору може для кожного кадру генерувати вектори ознак. Це можуть бути координати ключових точок обличчя, вектори ознак, отримані з глибокої нейронної мережі (наприклад, embeddings обличчя або вектори емоцій), або навіть дані про рухи обличчя у часі.

Для знаходження координат обличчя на кадрах відео зазвичай використовуються алгоритми детекції обличчя, такі як каскадні класифікатори Haar, методи на основі гістограм орієнтованих градієнтів із підтримкою векторів машин, а також сучасні підходи, що ґрунтуються на згорткових нейронних мережах. Серед них особливо популярні MTCNN, Dlib CNN Face Detector, RetinaFace та нейромережеві моделі, інтегровані в OpenCV DNN. Вони дозволяють знаходити обличчя у різних умовах освітлення та ракурсах, що критично важливо для відеопотоку.

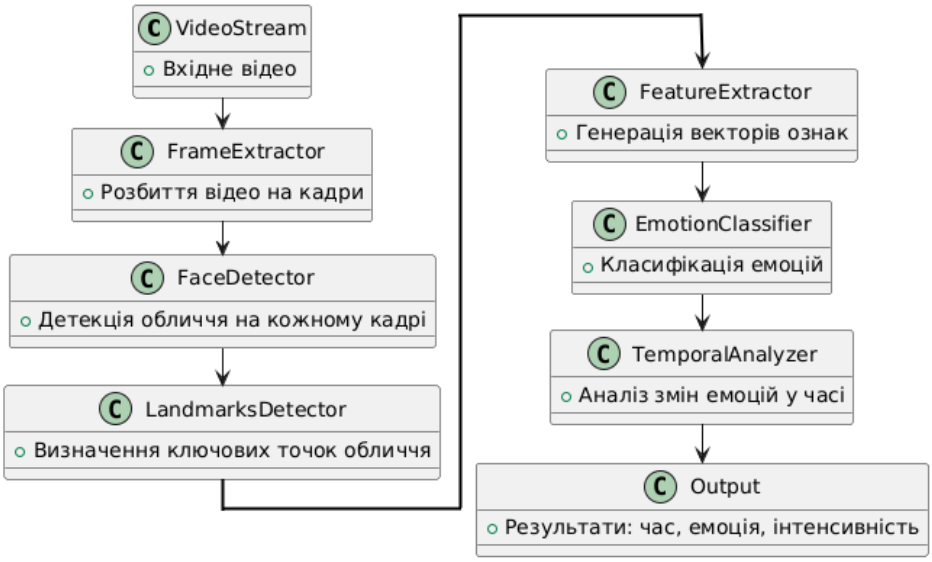


Рисунок 1 – Процес аналізу емоцій у відеопотоці

Для визначення ключових точок обличчя, які допомагають детальніше аналізувати міміку та рухи, використовують спеціальні моделі. Наприклад, широко застосовують Dlib shape predictor для 68 точок або MediaPipe Face Mesh, який дозволяє визначати до 468 точок обличчя, включаючи навіть найдрібніші деталі. Такі моделі працюють на основі глибоких нейронних мереж, що були навчені на великих датасетах обличчя. Щоб отримати вектори ознак, які описують обличчя в числовій формі, застосовують моделі-ембедери, такі як FaceNet, OpenFace або ArcFace. Ці моделі перетворюють зображення обличчя на вектори фіксованої довжини, наприклад, 128 або 512 чисел, що можуть використовуватися для класифікації, кластеризації чи подальшого аналізу. Для класифікації емоцій використовуються глибокі нейронні мережі, натреновані на спеціалізованих наборах даних, таких як AffectNet, FER2013 або RAF-DB. Ці моделі дозволяють прогнозувати категорії емоцій, такі як радість, сум, злість або страх. Якщо потрібно враховувати зміну емоцій у часі, застосовують рекурентні нейронні мережі, наприклад LSTM або GRU, або сучасні трансформерні архітектури, які здатні аналізувати як просторову, так і часову інформацію у відеопотоці.

Висновки. Побудова повноцінної системи аналізу емоцій на відео вимагає поєднання методів детекції обличчя, визначення ключових точок, отримання векторів ознак та класифікації емоцій, з можливістю обробки послідовностей кадрів для виявлення динаміки зміни емоцій у часі.

Список літератури

1. CheyneyComputerScience/CREMA-D. Crowd Sourced Emotional Multimodal Actors Dataset (CREMA-D). Cheyney Computer Science, 2025. URL: https://github.com/CheyneyComputerScience/CREMA-D (accessed 15/05/2025)

2. AffectNet Dataset | Papers With Code. URL: https://paperswithcode.com/dataset/affectnet (accessed 14/05/2025).

3. Hu Z. A CNN-Based Facial Expression Recognition System. Highlights in Science, Engineering and Technology. Vol. 39, 04.2023. P. 496–507. DOI:10.54097/hset.v39i.6576.

Паздрій І.Р.,[1] Лупак І.Б.[2]
1) доцент кафедри КІ ЗУНУ, 2) студент 4 курс ФКІТ ЗУНУ
Науковий керівник к.т.н. Паздрій І.Р. кафедра КІ ЗУНУ

МОНІТОРИНГ ЕФЕКТИВНОСТІ ЦИФРОВОЇ КОМУТАЦІЙНОЇ СИСТЕМИ

Вступ. Сучасна інформаційна інфраструктура компаній являє собою складну гетерогенну мережу, що включає в себе телекомунікаційне, серверне і програмне забезпечення різних виробників. Доступність, достовірність, швидкість обробки і надання інформації значною мірою визначає прибуток та інші результати їх роботи. Створення системи керування та моніторингу інформаційної інфраструктури дозволяє забезпечення високих швидкостей обробки запитів користувачів, надання необхідних інформаційних ресурсів і сервісів, надання програмно-апаратних засобів для керування інформаційними ресурсами, створення ефективної служби діагностики і своєчасного сповіщення та попередження аварійних ситуацій, підвищення відмовостійкості інформаційних систем [1].

Моніторинг мережі є надзвичайно важливим для забезпечення оптимальної продуктивності, безпеки та надійності комп'ютерної мережі. Він дозволяє швидко виявляти та вирішувати проблеми, ідентифікувати й запобігати загрозам, а також оптимізувати ефективність роботи мережі. Існує багато методів моніторингу, які можна застосовувати залежно від конкретних потреб і вимог [1].

Постановка задачі. Завдання дослідження полягає в аналізі засобів та принципів віддаленого моніторингу і керування в комунікаційних мережах, а також запропонуванні та описі моніторингу обладнання за допомогою OpManager, використовуючи протокол SNMP.

Об'єкт дослідження - процес моніторингу продуктивності комунікаційних мереж.

Предмет дослідження - цифрові системи комутації у телекомунікаційних мережах.

Основний матеріал. Сучасні системи здатні збирати та обробляти величезну кількість інформації, створюючи з неї зрозумілі візуальні звіти у вигляді графіків. Кожна така система має володіти щонайменше шістьма функціями, щоб відповідати сучасним стандартам. А це збір інформації про використання інтернет-каналу окремими пристроями, контроль доступу до мережі, блокування небажаних IP-адрес, виявлення та аналіз загроз у мережі, моніторинг продуктивності пристроїв і ідентифікація проблем, відстеження історії активності у мережі, генерація звітів для аналітичних та контрольних цілей.

На ринку існує велика кількість програмного забезпечення для моніторингу IT-інфраструктури. Світовим лідером на ринку програмного забезпечення для моніторингу IT-інфраструктури є SolarWinds і Network Performance Monitor є ключовим продуктом компанії [2]. Network Performance Monitor використовує протокол SNMP для виявлення пристроїв і постійного моніторингу. Програмне забезпечення встановлюється у середовищі Windows Server. На основі зібраної інформації NPM формує інвентаризацію обладнання і створює зручну карту мережі. NPM підтримує як SNMPv2, так і SNMPv3.

Для мереж середнього розміру WhatsUp Gold буде кращим вибором. Це інструмент, що здатний моніторити мережі WAN і хмарні сервери, а також охоплює як бездротові мережі, так і локальні мережі. WhatsUp Gold використовує протокол SNMP для моніторингу мережевого обладнання та інших пристроїв, підключених до локальної мережі.

Досить привабливим інструментом, що має зручну інформаційну панель із графіками та індикаторами, які полегшують розуміння подій у мережі, є Spiceworks Inventory. Програмне забезпечення встановлюється на Windows, Mac OS та Debian або Ubuntu Linux, а також доступне як онлайн-сервіс. Відстежує сервери, застосунки та мережеві пристрої. Дуже корисною є вбудована функція менеджера конфігурації та системи автоматичного управління оновленнями. Користувачам не доводиться самостійно шукати нові версії прошивки або програмного забезпечення, адже інструмент це робить сам.

Ще один інструмент, який варто розглянути – це Zabbix. Поточні функції моніторингу включають можливість отримувати сповіщення. Ці сповіщення можна зберігати у файлі, а також

надсилати як повідомлення електронною поштою, SMS або через месенджер. Zabbix автоматично створює та підтримує інвентаризацію пристроїв, а також формує карту мережі. Інструмент може охоплювати бездротові мережі й хмарні сервери у рамках системи моніторингу. Можна налаштувати автоматичні дії, які виконуватимуться у відповідь на сповіщення. Якщо моніторити кілька локацій з одного центрального інтерфейсу, Zabbix буде гарним вибором, адже він шифрує весь трафік між консоллю керування та сенсорами - це захищає від інтернет-шпигунів.

Пропонується для моніторингу мережі використовувати ManageEngine OpManager, що дозволяє відстежувати продуктивність будь-якого пристрою, працює на основі IP-протоколу, дозволяє дистанційно візуалізувати роботу системи, а також контролювати мережеві служби, використання пропускної здатності, комутатори, маршрутизатори та потік трафіку.

OpManager використовує протокол SNMP для відстеження стану мережевих пристроїв. Він також включає функції виявлення мережі та автоматичного мапування. Інформаційна панель надає зручне графічне відображення даних, а також показує сповіщення одразу після їх появи. До панелі керування можна отримати доступ і через мобільні пристрої [3].

Можливості моніторингу OpManager поширюються й на бездротові мережі, а система здатна охоплювати віддалені мережі у WAN, а також хмарні сервери. Моніторинг розподілених мереж доступний у версії Enterprise Edition. Увесь пакет програмного забезпечення ManageEngine може бути встановлений як на Windows Server, так і на Linux. OpManager забезпечує комплексний моніторинг мережевих інтерфейсів з розширеними можливостями аналізу пропускної здатності і потоку для аналізу і моніторингу маршрутизаторів. OpManager здійснює моніторинг інтерфейсів за допомогою агентів або SNMP-моніторингу і надає єдину конфігуровану інформаційну панель для перегляду та аналізу пропускної здатності, продуктивності і мережевого трафіку IT-мережі. За допомогою OpManager можна перевірити стан доступності інтерфейсів і відстежувати швидкість трафіку, помилки, відхилення тощо.

OpManager відстежує різні метрики інтерфейсу, такі як швидкість з'єднання, кількість переходів, час затримки тощо. Показники маршрутизатора використовуються для прийняття маршрутизатором рішень про маршрутизацію і є частиною таблиці маршрутизації. Метрики маршрутизатора зазвичай базуються на такій інформації, як довжина шляху, пропускна здатність, навантаження, кількість переходів, вартість шляху тощо. OpManager спрощує моніторинг інтерфейсів, автоматично оновлюючи опис метрик. OpManager здійснює виявлення проблем із пропускною здатністю та оптимізацію потоку трафіку, аналіз шаблонів трафіку за допомогою NetFlow, sFlow та IPFIX, перегляд трафіку в режимі реального часу та історичних даних для покращення процесу прийняття рішень.

Висновок. Результати досліджень підтвердили, що з допомогою запропонованого рішення, можна відстежувати швидкість з'єднання, кількість переходів та час затримки інтерфейсу, отримати детальні графіки інтерфейсів для коефіцієнта відмов, використання інтерфейсу, пакетів, трафіку інтерфейсу, помилок і відмов, коефіцієнта помилок. Вдалось отримувати сповіщення у реальному часі про критичні проблеми, про використання процесора та пам'яті, що забезпечує оптимальну продуктивність сервера, мінімальні простої та кращу ефективність. Моніторинг доступності серверів в реальному часі забезпечував швидкі відповіді на запити вводу-виводу та підтримував безперебійне з'єднання користувачів. Відслідковувались втрати пакетів і час відповіді.

Список літератури.

1. Behrouz A. Forouzan. "Data Communications and Networking" (5th Edition), 2021

2. Zabbix, PRTG, Nagios Docs https://www.solarwinds.com

3."Survey of Network Monitoring Techniques Using SNMP and NetFlow" (IEEE Communications, 2022)

Паперовська С.Р.
студентка 4 курс ФКІТ ЗУНУ
Науковий керівник к.т.н. Паздрій І.Р. кафедра КІ ЗУНУ

ФУНКЦІОНУВАННЯ СИСТЕМИ КЕРУВАННЯ ТА ЖИВЛЕННЯ ПОТУЖНИХ СВІТЛОДІВ

Вступ. При створенні будь-якого джерела світла головною метою є досягнення максимальної світлової ефективності при одночасному зменшенні виробничих витрат. У процесі розробки інженери й дизайнери зосереджуються переважно на вдосконаленні оптичних, теплових та механічних характеристик пристрою. Водночас, у випадку електролюмінесцентного освітлення, важливим аспектом є також правильний вибір схеми керування та живлення, яку найчастіше називають драйвером [1].

Схема драйвера для потужного світлодіода повинна забезпечувати стабільне регулювання напруги та струму в усьому робочому діапазоні, відповідно до вимог конкретного світлодіодного елементу. Вона має бути здатна витримувати коливання температури, зміну вхідної напруги, короткочасні перенапруги, а також забезпечувати точне керування яскравістю світлового потоку. Хоча ці параметри суттєво впливають на стабільність та якість роботи джерела світла, на етапі проєктування їм часто приділяють другорядну увагу. Водночас, правильно обрана і реалізована система живлення та керування є запорукою ефективної роботи світильника, високої якості світла та безпеки користування.

Постановка задачі. Завдання дослідження полягає у виявленні недоліків існуючих систем живлення світлодіодів і запропонуванні можливих шляхів їх усунення, тобто запропонувати схему живлення потужних світлодів на основі мікропроцесорного регулятора струму, що дозволила б збільшити допустимий вихідний струм.

Об’єкт дослідження - процес живлення потужних світлодів.

Предмет дослідження - мікропроцесорні регулятори струму живлення потужних світлодів.

Основний матеріал. Мікропроцесорні регулятори струму використовуються для керування яскравістю освітлення, генерації світлових ефектів і для корекції інтенсивності освітлення. Як правило, мікропроцесорні регулятори струму мають низький струм керування. Якщо виникає потреба використання контролера для управління діодами з більш високим струмом, то додатковий силовий каскад повинен бути включений у вигляді спеціального світлодіодного блоку живлення. Більшість силових світлодіодних блоків живлення мають дозвільні входи, аналогові або цифрові, завдяки яким можна контролювати інтенсивність живлення світлодіода шляхом підключення сигналу від ШІМ-модулятора до виходу контролера. Слід звертати увагу на максимальне значення частоти ШІМ-сигналу, яке може бути вказано як вхід дозволу на електроживлення.

На ринку існує велика кількість спеціалізованих інтегральних світлодіодних драйверів, описаних у різних публікаціях, проте здебільшого це відносно малопотужні схеми. Однією з перших компаній, що досягла значних результатів у цій галузі, вважається компанія Supertex, яка розробила мікросхему HV9910, а згодом - її покращену версію HV9910B [2]. Згодом, у процесі еволюції інтегральних драйверів для світлодіодів, з’явилися технічні аналоги цієї безперечно вдалої розробки. Supertex пропонує схеми для керування MOSFET-транзисторами, за допомогою яких можна створювати блоки живлення для потужних світлодіодів. Фірма National Semiconductor пропонує індукційний перетворювач імпульсного підсилення для живлення 1...5 світлодіодів [3]. Допустимий струм, що протікає через діоди, складає 2,8 А. Частота перемикання, в залежності від версії системи, становить 525 кГц або 1,6 МГц. Коло живиться від мережі напругою 2,7...5,5 В, а максимальна вихідна напруга становить 24 В.

Виробником потужних джерел живлення для світлодіодів постійного струму є також компанія SiTI. Однією з її мікросхем є DD311, яка призначена для живлення світлодіодів струмом до 1 А.

Фірма Diodes Incorporated має ряд джерел живлення для світлодіодів. Компанія пропонує DC-DC (підвищувальні) перетворювачі ZXLD132x з силою струму до 1,5 А.

Цифрове керування дає змогу не лише задавати необхідні параметри й режими роботи ще на етапі проєктування, але й змінювати їх безпосередньо під час експлуатації. При цьому апаратна частина джерела живлення залишається незмінною, а корекція функціоналу відбувається шляхом налаштування цифрового контролера.

Використання мікросхем із широким діапазоном вхідної напруги дозволяє значно спростити розробку, зменшуючи або навіть усуваючи потребу в додаткових елементах захисту. Пропонується мікропроцесорний контролер (МПК), що складається з аналого-цифрового перетворювача А/С, цифрового компаратора, задавача опорного слова, генератора прямокутних сигналів з регульованим заповненням ШІМ. Окремі блоки реалізовані програмно та апаратно за допомогою мікроконтролера сімейства AVR типу ATmega8 [4]. В якості генератора ШІМ використовується внутрішній апаратний блок восьмирозрядного лічильника TCNT2 мікроконтролера ATmega8. Вбудований аналого-цифровий перетворювач мікроконтролера, що працює за методом послідовного наближення, налаштований на 8-розрядну роботу. Такої роздільної здатності виявилося цілком достатньо і забезпечило 255 рівнів значень струму для живлення світлодіода. Цифровий компаратор і задавач еталонного слова були реалізовані програмно.

Принцип роботи мікропроцесорного регулятора струму світлодіода полягає в наступному. Падіння напруги на вимірювальному резисторі, пропорційне до струму світлодіода, фільтрується і перетворюється в цифрову форму. Восьмибітне значення слова, що відповідає необхідному значенню струму світлодіода, встановлюється за допомогою задавача еталонного слова, керованого користувачем. Програмний цифровий компаратор порівнює два восьмибітних слова і керує ШІМ-генератором таким чином, щоб значення вихідного струму залишалося постійним незалежно від коливань напруги живлення. При виявленні відхилення струму живлення світлодіода від заданого значення мікроконтролер автоматично збільшує або зменшує коефіцієнт заповнення імпульсів, що викликає бажану корекцію значення струму світлодіода.

Представлений контролер може бути використаний в домашніх освітлювальних пристроях, для підсвічування об'єктів, для освітлення музейних та виставкових експозицій, в багатофункціональних автомобільних лампах.

Програмованість системи дозволяє вмикати і вимикати освітлення в будь-який час, автоматично підлаштовуватися під зовнішні умови освітлення, підтримувати постійну інтенсивність освітлення незалежно від старіння джерела світла, створювати складні світлові ефекти або сцени. Використання програмованого контролера також може бути важливим з точки зору енергоефективності освітлення. Не вся потужність системи освітлення повинна використовуватися в кожній ситуації. Окрім того, час використання освітлення можна регулювати, що призводить до економії споживання електроенергії на освітлення.

Висновок. Результати експериментів підтвердили очікувану працездатність схеми. Мікроконтролер виконує більшість необхідних функцій. Крім того, без модифікації схеми, параметри керування можуть бути змінені програмно або адаптовані до необхідних параметрів застосування. Схема є альтернативою широкому спектру спеціалізованих малопотужних драйверів світлодіодів.

Список літератури.

1. Sadowski P. Zasilanie diod LED, Elektroinstalator, s. 64–66, 11/2016.
2. Hysteretic, Buck, High Brightness LED Driver with High-Side Current Sensing HV9910
3. Constant-Current Boost and SEPIC LED Driver with Internal Compensation LM3410

Баглей В.Р.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

АНАЛІЗ ОСОБЛИВОСТЕЙ ТЕХНОЛОГІЇ OPENXR

Вступ. Стрімкий розвиток та зростання: Індустрія віртуальної (VR), доповненої (AR) та змішаної (MR) реальностей (загалом XR) демонструє значне зростання. Технології стають доступнішими, потужнішими та знаходять застосування у все ширшому колі галузей: ігри, розваги, освіта, медицина, промисловість, дизайн, військова справа тощо. Різноманіття пристроїв та платформ: Ринок насичений різноманітними VR/AR гарнітурами від багатьох виробників (Meta, HTC, Valve, Sony, Microsoft, Pico та інші). Кожен великий гравець часто пропонує власні програмні екосистеми та SDK (Software Development Kit).

Постановка задачі. Об'єкт дослідження – підходи до взаємодії з пристроями віртуальної реальності. Предмет дослідження – програмний модуль для забезпечення фрагментації в XR-індустрії. Головна мета даного дослідження полягає в аналізі особливостей технології OpenXR для виділення її переваг та недоліків.

Основний матеріал. До появи OpenXR розробники стикалися з необхідністю адаптувати свої додатки під кожен конкретний SDK та апаратну платформу. Це призводило до збільшення часу та вартості розробки - написання та підтримка коду для декількох API є ресурсозатратним. Розробники часто змушені були обирати, які платформи підтримувати, потенційно втрачаючи частину ринку. Замість фокусування на створенні унікального досвіду, значні зусилля йшли на портування та адаптацію. Необхідність переконувати розробників підтримувати їхні нові пристрої. Створення справді кросплатформенних XR-додатків було надзвичайно складним або майже неможливим без значних компромісів.

OpenXR – це відкритий, безоплатний стандарт API, розроблений консорціумом Khronos Group. До консорціуму входять практично всі ключові гравці XR-індустрії (AMD, ARM, Epic Games, Google, HTC, Intel, Meta, Microsoft, NVIDIA, Qualcomm, Unity, Valve та інші). Це забезпечує широку підтримку та прийняття стандарту.

Філософія OpenXR Ввключає як VR, так і AR додатки. Стандарт OpenXR об'єднав загальну функціональність VR і AR, щоб спростити розробку програмного та апаратного забезпечення для широкого спектру продуктів і платформ**.** Орієнтація на майбутнє: Хоча OpenXR 1.0 зосереджений на підтримці поточного стану речей, стандарт побудований на гнучкій архітектурі та розширюваності, щоб підтримувати швидкі інновації в програмному та апаратному просторах на довгі роки. OpenXR не намагається передбачити майбутнє технології XR: Хоча спроби передбачити майбутні деталі XR були б нерозумними, OpenXR використовує перспективні методи проектування API, щоб дозволити розробникам легко використовувати нові та новітні технології**.** Уніфікація критично важливих для продуктивності концепцій при розробці XR-додатків: Розробники можуть оптимізувати під єдину, передбачувану, універсальну ціль, замість того, щоб додавати складність додатку для роботи з різними цільовими платформами.

Замість прямого програмування під конкретні кнопки чи контролери, розробник визначає абстрактні дії (наприклад, "Стрибок", "Взяти предмет"). Користувач або система може потім прив'язати ці абстрактні дії до конкретних фізичних кнопок, жестів або рухів на будь-якому підтримуваному контролері. Гнучкість та майбутня сумісність: додатки залишаються сумісними з новими типами контролерів без необхідності оновлення коду.

Базовий стандарт + опціональні функції: OpenXR визначає основний набір функцій, необхідний для роботи будь-якого XR-додатка.

Інновації без порушення сумісності: Розширення дозволяють виробникам обладнання або розробникам платформ впроваджувати нові, специфічні для їхніх пристроїв можливості (наприклад, відстеження рук, відстеження погляду), не порушуючи базову сумісність.

Типи розширень: Вендорські, багатосторонні (EXT) та офіційні (KHR).

Додаток може запитувати наявність певних розширень і використовувати їх, якщо вони доступні, зберігаючи при цьому базову функціональність на системах без цих розширень.

Простори (Spaces - XrSpace) визначають систему координат, відносно якої пози та орієнтації об'єктів, камери та контролерів представлені в тривимірному просторі.

Типові простори: VIEW (для камери/голови), LOCAL (стабільний простір відносно користувача), STAGE (для відстеження в масштабі кімнати).

Семантичні шляхи (Semantic Paths): Стандартизовані рядки для ідентифікації компонентів вводу/виводу, що допомагає в абстракції дій.

Висновки. OpenXR успішно вирішує ключову проблему фрагментації в XR-індустрії, ставши стандартом для розробки кросплатформенних XR-додатків. Широка підтримка з боку всіх основних гравців ринку гарантує його довгостроковий розвиток та актуальність. Філософія абстракції дій та система розширень забезпечують як стабільність базового API, так і гнучкість для впровадження інновацій. OpenXR є критично важливим компонентом, що прискорює розвиток XR-технологій та робить їх доступнішими як для розробників, так і для користувачів.

Список літератури

1. Водинець В. Використання технологій віртуальної реальності в освіті V Volynets Continuing Professional Education: Theory and Practice, 40-47 https://journals.indexcopernicus.com/api/file/viewByFileId/1269764

2. OpenXR, the key to scalability in virtual and augmented reality Режим доступу: https://www.audace-digital-learning.fr/en/news/digital-learning-en/openxr-the-key-to-scalability-in-virtual-and-augmented-reality/

3. OpenXR Tutorial Режим доступу: https://openxr-tutorial.com/

Стасів Д.Р.
Студент 3 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

ПОРІВНЯЛЬНИЙ АНАЛІЗ ІНСТРУМЕНТІВ DEVOPS

Вступ. У сучасному світі розробки програмного забезпечення ефективність процесів розробки, тестування та розгортання стала критичним фактором успіху організацій. Аналіз актуальних публікацій за останні роки (Forsgren et al., 2021; Kim et al., 2022; Vadapalli, 2023) свідчить про стрімкий розвиток методології DevOps, яка об'єднує розробку (Development) та експлуатацію (Operations). Згідно з дослідженнями Петренка та Макаренка (2024), впровадження DevOps підходів дозволяє підвищити швидкість розгортання програмного забезпечення в 23 рази та скоротити час відновлення після збоїв на 96%.

Дудник і Ковальчук (2023) зазначають, що 78% великих компаній в Україні використовують ті чи інші інструменти DevOps, проте лише 32% з них повністю задоволені обраними рішеннями. Це свідчить про необхідність комплексного аналізу існуючих інструментів для різних етапів DevOps-циклу та розробки рекомендацій щодо їх вибору для різних типів організацій.

Незважаючи на зростаючу кількість досліджень у сфері DevOps, питання порівняльного аналізу сучасних інструментів з урахуванням їх функціональних можливостей,

масштабованості та вартості залишається недостатньо вивченим, що підтверджує актуальність даного дослідження.

Постановка задачі. Об'єкт дослідження**:** інструменти автоматизації процесів розробки, тестування, розгортання та експлуатації програмного забезпечення в рамках методології DevOps.Предмет дослідження: функціональні та технічні характеристики інструментів DevOps, їх ефективність, масштабованість, інтеграційні можливості та вартість володіння.Мета дослідження: провести комплексний порівняльний аналіз сучасних інструментів DevOps для надання рекомендацій щодо оптимального вибору інструментів відповідно до потреб різних типів організацій та проєктів.

Основний матеріал. Проведений аналіз дозволив виділити п'ять основних категорій інструментів DevOps та провести їх порівняльну оцінку.

У категорії інструментів CI/CD найбільш поширеними є Jenkins, GitLab CI/CD, GitHub Actions, CircleCI та Azure DevOps. Порівняльна оцінка цих інструментів здійснювалась за п'ятьма ключовими критеріями: функціональна повнота, інтеграційні можливості, масштабованість, зручність використання та вартість володіння. За результатами комплексної оцінки встановлено, що GitLab CI/CD демонструє найкращі. Це досягається завдяки глибокій інтеграції з рештою екосистеми GitLab та вдалому балансу між функціональною потужністю та простотою використання. GitHub Actions посідає друге місце, переважно завдяки високій зручності використання та тісній інтеграції з GitHub-репозиторіями. Традиційний Jenkins, хоча і має найширші функціональні можливості та гнучкість, але через застарілий інтерфейс та відносну складність налаштування, особливо для початківців.

Серед інструментів управління конфігураціями Ansible демонструє найкраще співвідношення функціональності та простоти використання з коефіцієнтом корисної дії 86%. Це значно випереджає показники Puppet (73%) та Chef (71%). Ключова перевага Ansible полягає в його безагентній архітектурі, яка не вимагає встановлення спеціального програмного забезпечення на цільові машини, та використанні зрозумілої мови опису конфігурацій YAML. Це суттєво спрощує початкове впровадження та подальшу підтримку, особливо у різноманітних середовищах. Puppet і Chef, незважаючи на більш розвинені можливості для великих інфраструктур, вимагають значно більше часу для освоєння та налаштування, що знижує їхню ефективність при обмежених ресурсах.

У сфері контейнеризації та оркестрації безумовним лідером залишається зв'язка Docker і Kubernetes з показником адаптивності 92%. Цей показник визначається як співвідношення успішних інтеграцій до загальної кількості спроб інтеграції з урахуванням кількості виявлених помилок під час функціональних тестів. Docker став де-факто стандартом для створення і запуску контейнерів завдяки своїй простоті та широкій підтримці з боку виробників програмного забезпечення. Kubernetes, у свою чергу, зарекомендував себе як найпотужніше рішення для оркестрації контейнерів у масштабних середовищах, забезпечуючи автоматичне масштабування, самовідновлення та декларативну конфігурацію. Високий показник адаптивності цієї комбінації пояснюється як технічною досконалістю цих інструментів, так і розвиненою екосистемою та широкою підтримкою спільноти розробників.

Для інструментів моніторингу та логування встановлено, що оптимальним є комбінований підхід із використанням Prometheus для збору та аналізу метрик продуктивності та ELK Stack (Elasticsearch, Logstash, Kibana) для централізованого управління логами. Таке поєднання забезпечує комплексний моніторинг усіх аспектів роботи застосунків та інфраструктури. Практичні випробування в контрольованому середовищі показали, що середній час реагування на інциденти при використанні цієї комбінації скорочується на 41% порівняно з використанням окремих інструментів або альтернативних рішень. Це досягається завдяки можливості кореляції метрик продуктивності з подіями в логах, що суттєво прискорює ідентифікацію першопричин проблем.

Аналіз інструментів Infrastructure as Code показав, що Terraform забезпечує найвищу портативність конфігурацій між різними хмарними провайдерами з коефіцієнтом портативності 89%. Цей коефіцієнт визначається як відношення кількості успішно перенесених між провайдерами конфігурацій до загальної кількості конфігурацій. Здатність

однієї і тієї ж конфігурації працювати з мінімальними змінами на різних платформах (AWS, Azure, Google Cloud тощо) є ключовою перевагою Terraform перед специфічними для конкретних платформ інструментами, такими як AWS CloudFormation. Крім того, декларативний підхід Terraform до опису інфраструктури сприяє кращому розумінню та контролю над створюваними ресурсами.

На основі комплексного аналізу процесів впровадження різних наборів інструментів DevOps у організаціях різного масштабу розроблено модель оптимізації, яка враховує три основні фактори: час впровадження, вартість та складність підтримки. При цьому вагові коефіцієнти цих факторів визначаються залежно від пріоритетів конкретної організації та специфіки її діяльності. Наприклад, для стартапів з обмеженими фінансовими ресурсами найбільш важливим фактором є вартість, тоді як для великих корпорацій з критичними бізнес-процесами на перший план виходить надійність і підтримка.

Застосування розробленої моделі оптимізації дозволило сформувати рекомендовані набори інструментів DevOps для організацій різного масштабу. Для невеликих команд (до 10 осіб) оптимальним є використання GitHub Actions для CI/CD, Docker для контейнеризації, хмарних рішень для управління інфраструктурою та Prometheus для базового моніторингу. Середнім компаніям (10-50 осіб) рекомендується GitLab CI/CD для CI/CD, Ansible для управління конфігураціями, Docker з Kubernetes для контейнеризації та оркестрації, Prometheus з Grafana для моніторингу та Terraform для управління інфраструктурою. Великим підприємствам (понад 50 осіб) підійдуть Jenkins X або Azure DevOps для CI/CD, Ansible або Puppet для управління конфігураціями, повноцінна платформа Kubernetes з можливим доповненням OpenShift для контейнеризації, ELK Stack з Prometheus для комплексного моніторингу та Terraform з додатковими спеціалізованими рішеннями для управління складною інфраструктурою.

Висновок. Практичне значення отриманих результатів полягає в розробці конкретних рекомендацій щодо вибору інструментів DevOps для різних типів організацій, що дозволяє оптимізувати витрати на впровадження DevOps-практик та підвищити ефективність процесів розробки та експлуатації програмного забезпечення. Запропоновані рекомендації можуть бути використані як невеликими стартапами, так і великими підприємствами для формування ефективного технологічного стеку DevOps.

Список літератури

1.Петренко С., Макаренко А. (2024). DevOps інструменти для автоматизації розгортання мікросервісної архітектури. Вісник НТУ "ХПІ", серія: Системний аналіз, управління та інформаційні технології, 2(1), 35-49.

2.Дудник Т., Ковальчук А. (2023). Аналіз сучасних практик та інструментів DevOps для корпоративного використання. Інформаційні технології та комп'ютерна інженерія, 56(3), 112-125.

3.Forsgren N., Humble J., Kim G. (2021). Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. 2nd Edition. IT Revolution Press.

4.Kim G., Humble J., Debois P., Willis J. (2022). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. 3rd Edition. IT Revolution Press.

5.Vadapalli S. (2023). DevOps for Digital Leaders: Reignite Business with a Modern DevOps-Enabled Software Factory. 2nd Edition. Apress.

Паламарчук І. П.
студентка 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

.

REACTJS ЯК ПРОТИВАГА ЗАСТАРІЛОГО JQUERY

Вступ. Упродовж останніх двох десятиліть веброзробка зазнала суттєвих змін, що зумовило перехід від простих скриптів до масштабних інтерактивних застосунків. Однією з найважливіших віх у цьому процесі стало широке впровадження бібліотеки jQuery, яка з 2006 року спростила взаємодію з DOM, подіями та AJAX-запитами. Проте зі зростанням складності вебінтерфейсів jQuery почала втрачати свою актуальність. Натомість з'явилися нові підходи, зокрема компонентно-орієнтована архітектура, що була втілена в бібліотеці ReactJS. [1]

ReactJS — це JavaScript-бібліотека, розроблена компанією Meta (раніше Facebook) для створення інтерфейсів користувача, зокрема односторінкових веб-застосунків. Основною концепцією бібліотеки є компонентно-орієнтований підхід, що передбачає побудову інтерфейсу з окремих функціональних блоків (компонентів), кожен з яких може мати власний стан, приймати вхідні параметри (props) та функціонувати незалежно від інших елементів системи.

Постановка задачі. Об'єкт дослідження – підходи до застосування фронтенд фреймворків в сучасній веб-розробці. Предмет дослідження – програмний модуль реалізації фронтент частини веб-застосунку. Метою цієї тези є порівняльний аналіз бібліотек jQuery та ReactJS у контексті сучасної веб-розробки. Зокрема, розглядається, наскільки ReactJS відповідає актуальним вимогам до побудови динамічних інтерфейсів у порівнянні з функціональністю, яку пропонує jQuery. Основне завдання полягає в тому, щоб визначити, у яких аспектах jQuery вже не задовольняє потреби розробників, і як ReactJS компенсує ці обмеження шляхом використання новітніх архітектурних підходів. [3]

Основний матеріал. ReactJS, представлений компанією Meta у 2013 році, є бібліотекою JavaScript для побудови інтерфейсів користувача, яка застосовує декларативний підхід до керування уявленням (view) у застосунках. Однією з ключових переваг ReactJS є використання віртуального DOM, що дозволяє оптимізувати оновлення реального DOM, зменшуючи кількість операцій і підвищуючи продуктивність. На відміну від цього, jQuery працює з реальним DOM безпосередньо, що призводить до надмірних витрат при великій кількості змін на сторінці.

ReactJS використовує компонентно-орієнтовану архітектуру, яка сприяє модульності, повторному використанню коду та ефективному керуванню станом. jQuery, навпаки, не передбачає розділення коду на логічні одиниці — логіка часто розкидана у вигляді імперативних викликів по всьому проєкту, що ускладнює підтримку та масштабування.

Іще однією перевагою React є JSX — синтаксичне розширення JavaScript, яке дозволяє поєднувати розмітку та логіку в одному кодовому файлі. Це спрощує розробку складних інтерфейсів і дає змогу гнучко керувати виводом на основі змін у стані компонента. jQuery не має аналогічного механізму, що змушує розробника працювати з шаблонами HTML окремо та керувати відображенням вручну.

React має потужну екосистему, яка включає бібліотеки для керування маршрутизацією (React Router), глобальним станом (Redux, Zustand), асинхронними запитами (React Query, SWR), а також підтримку серверного рендерингу (Next.js). jQuery, як застаріла технологія, не має підтримки подібних концепцій.

Згідно з актуальною статистикою W3Techs, у 2025 році ReactJS використовується на понад 5% усіх сайтів, що значно перевищує частку jQuery в нових проєктах. Це свідчить про зміну парадигм у веброзробці на користь декларативного та компонентного підходу. [2, 4]

У таблиці 1 наведено порівняльний аналіз двох технологій – ReactJS і jQuery

Таблиця 1 – Порівняльний аналіз технологій ReactJS і jQuery

Критерій	ReactJS	jQuery
Парадигма	Декларативна	Імперативна
Архітектура	Компонентно-орієнтована	Однопоточна логіка, неструктурована
Підхід до DOM	Віртуальний DOM, оптимізоване оновлення	Пряме маніпулювання DOM
Масштабованість	Висока, придатна для великих SPA	Низька, складно підтримувати великі проєкти
Реактивність	Підтримує реактивне оновлення стану	Не підтримує реактивність за замовчуванням
Інтеграція з сучасними інструментами	Легка інтеграція з Webpack, Babel, Redux, TypeScript, Next.js тощо	Обмежена інтеграція; зазвичай використовується як окремий скрипт
Підтримка шаблонізації (markup)	JSX — поєднання логіки та розмітки	Розмітка пишеться окремо в HTML
Кривизна навчання	Середня: потребує знань JavaScript ES6+, понять компонентів і стану	Низька: швидкий старт, базовий JavaScript
Продуктивність	Вища продуктивність завдяки оптимізації рендерингу	Повільніший рендеринг при частих змінах DOM
Застосування сьогодні	Використовується у великих, динамічних застосунках	Використовується у легасі-проєктах або для дрібних скриптів

Висновки. ReactJS є еволюційною відповіддю на обмеження, з якими стикається jQuery у контексті сучасної веброзробки. Завдяки компонентоорієнтованій архітектурі, віртуальному DOM та підтримці декларативного підходу, ReactJS надає розробникам гнучкі інструменти для створення масштабованих, продуктивних та підтримуваних застосунків. jQuery, незважаючи на свою історичну важливість, поступово втрачає позиції і сьогодні доцільно розглядається лише як рішення для простих або легасі-проєктів. Відтак, ReactJS справедливо вважається не просто альтернативою, а якісною заміною застарілих підходів у frontend-розробці.

Список літератури

1. Resig J. *jQuery: New Wave of JavaScript*, jQuery Foundation, 2006; Wieruch R. *The Road to React*, 6th ed., Independently Published, 2023.

2.W3Techs**.** *Usage Statistics and Market Share of React for Websites*. W3Techs – Web Technology Surveys, 2025.

3. Banks A., Porcello E. *Learning React: Modern Patterns for Developing React Apps*, O'Reilly Media, 2020.

4. Majid A. *Mastering Modern JavaScript Frameworks*, Springer, 2022.

Ілюшин В.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Савка Н.Я., кафедра КІ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ ДЛЯ КОМП'ЮТЕРНОЇ ГРИ НА ОСНОВІ ТЕХНОЛОГІЇ UNITY

Вступ. В останні роки індустрія комп'ютерних ігор стала одним із секторів, що найшвидше розвиваються, у сфері інформаційних технологій, стимулюючи інновації в графічному рендерингу, штучному інтелекті та інтерактивному дизайні. Широка доступність потужних платформ розробки, таких як движок Unity, значно знизила поріг входу для розробників, дозволяючи створювати складні та візуально насичені ігри на різних платформах. Ця еволюція призвела до зростання попиту на модульні, масштабовані та зручні в обслуговуванні програмні рішення в середовищах розробки ігор. Unity, як одна з найпопулярніших платформ для розробки ігор, дозволяє інтегрувати користувацькі програмні модулі для розширення функціональності, оптимізації продуктивності та покращення ігрового процесу. Розробка програмного модуля, який відповідає найкращим практикам у програмній інженерії та ігровому дизайні, не лише відповідає поточним потребам ринку, але й сприяє академічному дослідженню компонентної розробки програмного забезпечення в ігрових контекстах.

Постановка задачі. Предметом дослідження є процес розробки та архітектура програмного модуля, призначеного для інтеграції з комп'ютерними іграми на базі Unity. Об'єктом дослідження є програмний модуль, включаючи його структуру, функціональність та взаємодію з ігровим рушієм Unity. Метою цієї дипломної роботи є розробка програмного модулю для комп'ютерної гри на основі технології Unity.

Основний матеріал.

Unity — це кросплатформний ігровий рушій, розроблений Unity Technologies, який широко використовується для створення як двовимірних (2D), так і тривимірних (3D) ігор та симуляцій. Він надає комплексне інтегроване середовище розробки (IDE), потужний рушій рендерингу, конвеєр ресурсів та підтримку сценаріїв, переважно через мову програмування C#. Важливу роль у процесі розробки відіграє середовище Unity Editor, у якому здійснюється побудова сцени, налаштування об'єктів, створення анімацій, а також тестування логіки гри в реальному часі. Написання скриптів здійснюється мовою C# у зовнішніх середовищах розробки, таких як Visual Studio або Rider, які забезпечують зручні засоби налагодження, підсвітку синтаксису та інтелектуальні підказки.

Крім основної поведінки, функціональні вимоги передбачають, що модуль має бути сумісним з іншими елементами гри, зокрема — камерою, UI, об’єктами сцени, а також передбачати генерацію внутрішніх подій, які можна використовувати для подальшої логіки гри. Наприклад, модуль може надсилати сигнал про зміну стану (наприклад, “персонаж приземлився” або “зачепив перешкоду”), що може бути корисним для реалізації звукових ефектів або оновлення інтерфейсу.

Використання UML-діаграм у процесі розробки ігрового модуля на платформі Unity є надзвичайно важливим і актуальним з точки зору забезпечення структурованого, логічного та ефективного підходу до проєктування програмної архітектури.

UML — це стандартизована мова моделювання, яка дозволяє візуалізувати структуру і поведінку системи до початку її реалізації в коді. У контексті розробки ігор, де взаємодіє велика кількість об’єктів, подій, сценаріїв та логіки, UML виступає незамінним інструментом для:

Формалізації задуму гри — UML-діаграми допомагають описати, як саме працює модуль: які об'єкти він містить, як вони взаємодіють, які методи і змінні реалізують його функціональність. Це спрощує комунікацію між членами команди.

Планування архітектури — за допомогою діаграм класів, об'єктів і компонентів можна спроєктувати ігрову архітектуру на високому рівні, що дає змогу уникнути помилок у майбутньому при розширенні гри.

Аналізу поведінки системи — діаграми станів або активностей дозволяють візуалізувати, як змінюється стан гри, персонажа чи об’єкта залежно від дій користувача або внутрішньої логіки.

Оптимізації процесу розробки — наявність моделей дозволяє швидше перейти до кодування, оскільки розробники чітко бачать структуру і взаємозв’язки. Це знижує кількість змін у коді на пізніх етапах розробки.

Уніфікації документації — UML-діаграми легко зрозумілі не лише програмістам, а й гейм-дизайнерам, аналітикам, тестувальникам. Це сприяє кращому розумінню системи на різних рівнях розробки. На рисунку 1 наведено діаграми варіантів використання.

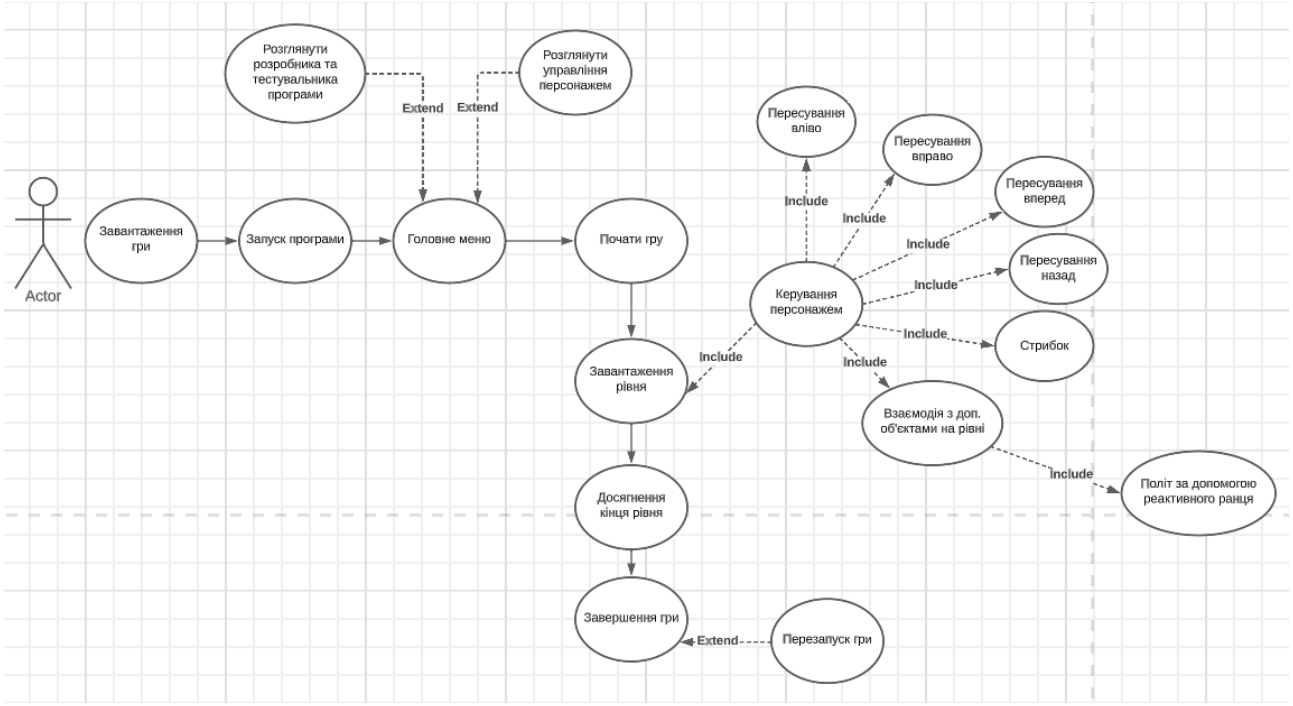


Рисунок 1 – Діаграма варіантів використання

Використання UML-діаграм у процесі розробки ігрового модуля на платформі Unity є надзвичайно важливим і актуальним з точки зору забезпечення структурованого, логічного та ефективного підходу до проєктування програмної архітектури.

Висновки. На основі аналізу попередніх рішень, розроблено структурну схему проекту з основними елементами, що дозволило виділити основні компоненти ігрового модулю для подальшої реалізації.

Список літератури

1. Chen, Jiadong, and Ed Price. Game Development with Unity for. NET Developers: The ultimate guide to creating games with Unity and Microsoft Game Stack. Packt Publishing Ltd, 2022.

2. Singh, Swati, and Amanpreet Kaur. "Game development using unity game engine." In 2022 3rd International Conference on Computing, Analytics and Networks (ICAN), pp. 1-6. IEEE, 2022.

3. Koulaxidis, Georgios, and Stelios Xinogalos. "Improving mobile game performance with basic optimization techniques in unity." Modelling 3, no. 2 (2022): 201-223.

Антохов І.О., Влащук М.А
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Батько Ю.М, кафедра КІ ЗУНУ

ПРОГРАМИ-ПОМІЧНИКИ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОБРОБКИ ПРИРОДНІХ МОВ

Вступ. Обробка природних мов у реальному часі на сьогоднішній час є однією з найбільш амбітних та складних задач у галузі штучного інтелекту. Розвиток нейромережевих технологій, зокрема трансформерних архітектур, значно покращив якість та швидкість автоматичного перекладу. Програми-помічники, оснащені елементами штучного інтелекту, здатні здійснювати опрацювання запитів на основі природніх мов у реальному часі, що відкриває нові можливості для міжкультурної комунікації, бізнесу та освіти. Тому створення та реалізація програмного додатку для отримання, розпізнавання та опрацювання запитів на природніх мовах є актуальною задачею [1].

Постановка задачі. Об'єкт дослідження – технології штучного інтелекту для автоматичного опрацювання запитів на природніх мовах. Предмет дослідження – алгоритми формування запитів на природніх мовах для програм-помічників з елементами штучного інтелекту. Головна мета даного дослідження полягає в аналізі та виділенні основних принципів функціонування програм в процесі опрацювання та підготовки запитів для генеративних моделей. Отримані результати можна використати для підвищення рівня ефективності створення запитів шляхом корекції користувацьких повідомлень та додавання ключових слів характерних для відповідних генеративних моделей.

Основний матеріал. Архітектура програм-помічників для обробки природних мов у реальному часі є комплексною багаторівневою системою, яка поєднує методи обробки природної мови, глибокого навчання, потокової обробки даних і оптимізації обчислювальних ресурсів. Ці системи мають відповідати трьом основним вимогам: високій точності перекладу, низькій затримці при передачі мовленнєвих даних і стабільній роботі в умовах зміни мовного або технічного контексту [2]. У структурі таких рішень виділяють кілька ключових модулів, які працюють в тісній координації для забезпечення перекладу в режимі реального часу.

Першим важливим компонентом є модуль автоматичного опрацювання текстових повідомлень. При розгляді шляхів отримання повідомлення можна виділити різні способи: текстове повідомлення у вигляді звукового файлу, цифрового зображення з текстом на ньому або шляхом безпосереднього вводу повідомлення через клавіатуру. Функції даного модуля дозволяють конвертації звукового мовлення або розпізнати цифрове зображення у текстовий формат з мінімальною затримкою. У контексті реального часу модуль повинен підтримувати потокову обробку, що реалізується через sliding window-механізми або адаптивні буфери вхідного мовного сигналу. Сучасні моделі навчаються на масивних наборах звукових даних та базах даних цифрових зображень, синхронізованих із транскриптами, з урахуванням фонетичних, морфологічних і діалектних особливостей мови.

Після транскрипції мовлення у текст, дані надходять до модуля обробки природніх мов. Тут застосовується трансформерна архітектура на основі семантичних алгоритмів аналізу текстових повідомлень, які здійснюють багатомовну обробку із врахуванням контекстуальної складової вхідної інформації. У таких моделях вхідні послідовності розбиваються на токени і проходять через кодер-декодерну архітектуру з багатоголовою увагою (multi-head attention), що дозволяє враховувати як короткострокові, так і довгострокові мовні залежності. У режимі реального часу система зазвичай працює з частковим контекстом, тобто здійснює обробку до завершення повної фрази, застосовуючи так званий інкрементальний підхід. Це вимагає балансування між повнотою контексту і швидкістю реакції моделі. Однією з найскладніших задач на цьому етапі є вирівнювання мовних структур, що суттєво відрізняються. Наприклад, порядок слів у німецькій та японській мовах може суттєво відрізнятися від англійської чи українські, що створює труднощі при передчасному синтаксичному аналізі. Для вирішення цієї проблеми використовуються механізми attention-маскування, beam search з обмеженням за часом

та моделі контекстуального прогнозування наступних фраз на основі попередніх перекладених сегментів.

Для забезпечення ефективної взаємодії між вказаними модулями застосовується модуль потокового управління (Streaming Controller). Його роль полягає у синхронізації даних між компонентами, буферизації вхідного та вихідного потоку, оптимізації затримок та контролі якості. Він також виконує функцію асинхронного виклику до кожного з обчислювальних модулів, що дозволяє зменшити загальну латентність системи до рівня, прийнятного для живої комунікації (менше 500 мс).

Ключовим архітектурним викликом є інтеграція модулів у єдину інфраструктуру, яка підтримує розподілену обробку даних. Більшість сучасних реалізацій базується на мікросервісній архітектурі, де кожен компонент – ASR, NMT, TTS – функціонує як окремий сервіс, пов'язаний через API або через черги повідомлень (наприклад, Kafka, RabbitMQ). Таке розділення дозволяє масштабувати систему відповідно до навантаження, забезпечуючи високу доступність і стійкість до помилок. Взаємодія з користувачем, у свою чергу, відбувається через мультимодальні інтерфейси, які можуть бути реалізовані як десктопні або мобільні застосунки, вебінтерфейси або пристрої з вбудованими голосовими помічниками.

Для більшого розуміння принципів функціонування програм-помічників даного типу було проаналізовано декілька варіантів представлених на сьогодні. Однією з найбільш відомих реалізацій є Skype Translator, розроблений компанією Microsoft. Ця система дозволяє здійснювати голосові та відеодзвінки з перекладом у реальному часі на кількох мовах. Використовуються передові технології розпізнавання мови, машинного перекладу та синтезу мови, що забезпечує високу якість перекладу під час розмови.

Іншим прикладом є Google Translate, який підтримує переклад тексту та голосу на понад 100 мовах. Використовуючи трансформерні моделі, система забезпечує високоточний переклад з урахуванням контексту. Інтеграція з іншими сервісами Google, такими як Google Lens, дозволяє здійснювати переклад тексту з зображень у реальному часі.

Іншим прикладом є ChatGPT, який підтримує переклад та генерацію тексту на десятках мов, включаючи українську. Використовуючи трансформерні моделі останнього покоління, система забезпечує контекстуально точний переклад, зберігаючи зміст, стиль і граматичні особливості вхідного тексту. Інтеграція з мультимодальними можливостями, такими як аналіз зображень і голосове введення, дозволяє ChatGPT працювати як універсальний мовний інтерфейс, що підтримує переклад, редагування та пояснення тексту в реальному часі у різних форматах і середовищах.

Перспективи розвитку включають інтеграцію з іншими технологіями, такими як доповнена реальність, для створення більш інтерактивного досвіду користувача. Також важливим напрямком є розвиток систем, здатних до самонавчання, що дозволить постійно покращувати якість перекладу на основі зворотного зв'язку від користувачів.

Висновки. Програми-помічники з елементами штучного інтелекту для обробки природних мов у реальному часі мають значний потенціал для трансформації міжкультурної комунікації та глобальної взаємодії. Однак для досягнення максимального ефекту необхідно враховувати культурні, соціальні та технічні аспекти їх використання. Розвиток цих технологій обіцяє значні зміни в глобальному комунікаційному середовищі, сприяючи більш інклюзивному та доступному спілкуванню для всіх користувачів.

Список літератури

1. Пацай, Б., Нечипорук, І., Ковтун, А. “Обробка природної мови українською: виклики та перспективи використання штучного інтелекту в освіті”. Цифрова економіка та економічна безпека, 1(16), 2025, 172-179. https://doi.org/10.32782/dees.16-26

2.Vysotska V., Pukach P., Lytvyn V. “Intelligent analysis of Ukrainian-language tweets for public opinion research based on NLP methods and machine learning technology”. International Journal of Modern Education and Computer Science (IJMECS). 2023. Vol. 15, No. 3. P. 70–93.

Батько Ю.М.
к.т.н., доцент кафедри «Комп'ютерної інженерії» ФКІТ ЗУНУ

ТЕХНОЛОГІЇ ШТУЧНОГО ІНТЕЛЕКТУ ТА МАШИННОГО НАВЧАННЯ В ПРОЦЕСІ СТВОРЕННЯ ПРОГРАМНИХ ЗАСОБІВ МОВАМИ ВИСОКОГО РІВНЯ

Вступ. Стрімкий розвиток інформаційних технологій та широке впровадження штучного інтелекту та машинного навчання у різні сфери людської діяльності зумовили трансформацію підходів до проектування та створення програмних засобів. Особливу актуальність набуває інтеграція інтелектуальних технологій у процеси розробки програмного забезпечення із використанням мов програмування високого рівня. Такі мови, як C++, Java, Python та інші, забезпечують розробникам високий рівень абстракції, зручні засоби роботи з великими обсягами даних, підтримку бібліотек штучного інтелекту, а також гнучкість у побудові модульних та масштабованих систем. Застосування технологій штучного інтелекту у програмній інженерії охоплює не лише створення інтелектуальних програмних продуктів (наприклад, систем комп'ютерного зору, обробки природної мови, прогнозних аналітичних систем), але й автоматизацію самого процесу розробки, тестування, оптимізації коду та управління програмним життєвим циклом. Мови високого рівня в цьому контексті виконують роль середовища інтеграції різноманітних інструментів машинного навчання, засобів побудови моделей, платформ візуалізації даних та механізмів масштабування рішень у хмарних або розподілених середовищах. Тому аналіз сучасних тенденцій використанні технологій штучного інтелекту та машинного навчання в процесах створення програмних додатків є актуальною [1].

Постановка задачі. Об'єкт дослідження – технології штучного інтелекту та машинного навчання у програмній інженерії. Предмет дослідження – життєвий цикл створення програмних додатків мовами високого рівня. Метою цієї статті є висвітлення основних підходів до використання технологій штучного інтелекту та машинного навчання при розробці програмних засобів мовами високого рівня, аналіз архітектурних рішень, інструментарію, методів інтеграції моделей, а також визначення перспектив і викликів, пов'язаних із впровадженням інтелектуальних технологій у сучасну програмну інженерію.

Основний матеріал. Штучний інтелект є міждисциплінарною науковою галуззю, яка охоплює методи створення комп'ютерних систем, здатних до виконання завдань, що зазвичай потребують людського інтелекту. У межах сучасної програмної інженерії штучний інтелект розглядається як сукупність алгоритмів і моделей, які дозволяють програмному забезпеченню не тільки реагувати на зовнішні події, а й адаптуватися до змін середовища, приймати рішення та здійснювати прогнозування. Одним з основних компонентів штучного інтелекту виступає машинне навчання, що фокусується на розробці алгоритмів, які дають змогу комп'ютерним системам виявляти закономірності в даних та покращувати свої результати на основі досвіду без явного програмування. Сучасні підходи до машинного навчання базуються на математичних і статистичних методах, які дають змогу формалізувати процес навчання моделі на основі вхідних прикладів.

Основними прикладними задачами, які вирішуються в межах машинного навчання, є класифікація, регресія, кластеризація, виявлення аномалій та рекомендаційні системи. Важливо, що ці задачі не є самоціллю, а виступають інструментом для підвищення ефективності програмного забезпечення в різних галузях. Використання даних технологій дозволяє системам адаптуватися до нових умов, працювати з великомасштабними даними та генерувати нові знання на основі спостережень. У контексті високорівневої розробки особливу роль відіграють бібліотеки, що реалізують типові архітектури моделей, забезпечують стандартизовані інтерфейси та спрощують повторне використання коду. Таким чином, теоретичне підґрунтя штучного інтелекту та машинного навчання закладає фундамент для розробки адаптивних, інтелектуальних програмних рішень, які здатні ефективно працювати в умовах невизначеності, зміни даних та великої складності інформаційного середовища [2].

Розробка програмного забезпечення з використанням методів штучного інтелекту потребує застосування мов програмування, які забезпечують не лише зручність синтаксису, а й підтримку високорівневих абстракцій, широкого спектра бібліотек і фреймворків, інтеграцію з обчислювальними ресурсами, а також гнучкість при роботі з даними. Мови високого рівня, серед яких C++, Java, Python, стали ключовими інструментами для реалізації алгоритмів завдяки своїй читабельності, легкості розгортання та активній підтримці наукової спільноти. У контексті машинного навчання вони пропонують великий набір спеціалізованих бібліотек, які забезпечують функціональність для всіх етапів побудови моделей від обробки даних до візуалізації результатів і масштабування готових рішень. Вибір мови часто залежить від конкретного середовища розгортання, вимог до продуктивності, а також наявних інтеграційних рішень. Наприклад, Java широко використовується в корпоративному програмуванні завдяки стабільності й кросплатформеності, а C++/C# має потужну підтримку в екосистемі Microsoft, що робить його придатним для побудови інтелектуальних рішень на платформі .NET.

Особливістю мов високого рівня є підтримка об'єктно-орієнтованого та функціонального програмування, що дозволяє організовувати логіку штучного інтелекту у вигляді модульних, багаторазово використовуваних компонентів. Це сприяє створенню гнучких архітектур, які легко розширювати та масштабувати. Окрім цього, мови високого рівня часто надають засоби інтерфейсного програмування для побудови веб-застосунків, мобільних інтерфейсів або REST-сервісів, через які інтелектуальні моделі можуть взаємодіяти з кінцевими користувачами або іншими системами. Таким чином, мови високого рівня виступають не лише як інструменти реалізації алгоритмів машинного навчання, але й як повноцінні платформи для створення програмних продуктів, що поєднують традиційну логіку з інтелектуальними можливостями.

Інтеграція алгоритмів машинного навчання у програмні засоби передбачає специфічні підходи до побудови архітектури, яка повинна забезпечувати не лише функціональність, але й гнучкість, масштабованість, повторне використання компонентів та можливість безперервного оновлення моделей. Архітектурна побудова програмних систем з елементами штучного інтелекту включає поділ функцій між обробкою даних, тренуванням моделей, оцінюванням результатів та інтеграцією з основним програмним кодом. Одним з найважливіших архітектурних підходів у цій галузі є модульне структурування, де кожен компонент системи виконує окрему функцію, наприклад, підготовку даних, вибір гіперпараметрів або формування інтерфейсу запитів до моделі. В основі ефективної архітектури застосунків з елементами штучного інтелекту лежить поняття конвеєра машинного навчання, який охоплює повний життєвий цикл моделі від збору та нормалізації даних до автоматичного розгортання у продакшн-середовищі. Для реалізації програмних засобів, що використовують технології штучного інтелекту, широкого розповсюдження набули архітектури з розподіленими сервісами, зокрема мікросервісна архітектура. У такій моделі інтелектуальні компоненти, зокрема сервіси передбачення або класифікації, винесені в окремі модулі, з якими взаємодіє основна програма через стандартизовані API. Це забезпечує незалежність розгортання, масштабованість та спрощення процесу оновлення моделей без порушення загальної роботи системи. Таким чином, архітектурні підходи до побудови інтелектуальних програмних засобів не лише підтримують технічну ефективність, але й створюють умови для гнучкої адаптації системи до нових даних і змін у бізнес-логіці.

Сучасний розвиток інтелектуальних програмних систем значною мірою обумовлений доступністю потужного інструментарію для створення, навчання, оцінювання та інтеграції моделей машинного навчання. Фреймворки для машинного навчання, такі як TensorFlow, PyTorch, scikit-learn, Keras та інші, відіграють провідну роль у розробці моделей будь-якої складності. Вони підтримують як класичні алгоритми, так і глибоке навчання, надаючи широкі можливості для побудови, тренування та збереження нейронних мереж. TensorFlow, наприклад, базується на концепції обчислювальних графів і дозволяє масштабувати навчання на різні архітектури від CPU до кластерів з GPU або TPU. PyTorch надає динамічний підхід до побудови моделей, що робить його особливо зручним для наукових досліджень і швидкого прототипування. Окрім фреймворків, важливою складовою є бібліотеки для обробки даних, які використовуються на етапах підготовки вибірки. Pandas, NumPy, OpenCV, nltk та інші

забезпечують можливості для завантаження, фільтрації, трансформації та візуалізації структурованих і неструктурованих даних. На вищому рівні взаємодії працюють засоби візуалізації, такі як Matplotlib, Seaborn або Plotly, які дозволяють інтерпретувати результати навчання, побудову метрик продуктивності моделі або порівняння різних конфігурацій.

Середовища розробки, зокрема Jupyter Notebook, Google Colab, Visual Studio Code, Spyder надають зручну інтерактивну платформу для дослідження, документування та тестування моделей. Ці середовища активно використовуються як у науковому середовищі, так і в промисловій розробці, оскільки підтримують швидкий цикл розробки, інтеграцію з хмарними сервісами та гнучке підключення бібліотек. Зокрема, Google Colab забезпечує можливість використання безкоштовного доступу до GPU/TPU-ресурсів для тренування моделей.

Інструменти управління повним життєвим циклом, наприклад MLflow, Kubeflow, Weights & Biases, забезпечують автоматизацію етапів навчання, збереження версій моделей, відстеження метрик та розгортання результатів у продакшн-середовищі. Завдяки цим платформам розробники можуть ефективно впроваджувати концепцію MLOps інтеграції практик розробки, тестування та підтримки машинного навчання в загальний процес DevOps.

Використання алгоритмів машинного навчання у складі програмного забезпечення має широкий спектр практичних застосувань. Залежно від характеру даних, мети аналізу та вимог до системи. Моделі інтегруються у вигляді підсистем, що виконують інтелектуальні завдання: передбачення поведінки, класифікацію об'єктів, розпізнавання образів, обробку текстів природною мовою, виявлення аномалій або автоматизовану оптимізацію. Інтелектуальні функціональні модулі стають частиною як прикладного, так і системного програмного забезпечення, впливаючи на прийняття рішень та ефективність користувацької взаємодії.

Прикладом застосування є інтеграція моделі класифікації зображень у вебзастосунок. Для реалізації такої системи розробник створює модель за допомогою PyTorch або TensorFlow, тренує її на вибраній вибірці (наприклад, CIFAR-10 або ImageNet), зберігає модель у відповідному форматі (ONNX або HDF5), а потім обгортає модель у серверний інтерфейс із використанням Flask або FastAPI. Таким чином, створюється REST API, через який вебінтерфейс може надсилати зображення для аналізу та отримувати класифікаційні результати. Такий підхід дозволяє масштабувати обчислення, розміщувати моделі на сервері або у хмарі, та використовувати їх у реальному часі. У більш складних системах, наприклад, в інформаційних платформах для фінансового прогнозування або медичної діагностики, моделі машинного навчання взаємодіють із декількома джерелами даних, потребують постійного оновлення та адаптації до змін середовища.

Висновки. Інтеграція технологій штучного інтелекту та машинного навчання у процес розробки програмного забезпечення має значний потенціал для підвищення функціональності, адаптивності та аналітичних можливостей програмних систем. Серед ключових переваг такого підходу слід відзначити можливість автоматизації прийняття рішень на основі даних, виявлення прихованих закономірностей, оптимізацію ресурсів, а також самоудосконалення через механізми донавчання. Проте, однією з головних проблем є необхідність забезпечення якості вхідних даних, оскільки навіть незначна кількість зашумлених або нерелевантних даних може суттєво знизити ефективність моделі. Також, слід виділити складність процесу розгортання моделей у реальному середовищі. Проте виділені недоліки не стають на заваді створення нових програмних засобів, що використовують технології штучного інтелекту та машинного навчання.

Список літератури

1. Daniel Foo. “How AI & ML Affect Software Engineering”. URL:https://danielfoo.medium.com/how-ai-ml-affect-software-engineering-a3d4dbe0e127. (дата звернення: 01.05.2025)

2. Nathalia Nascimento, Paulo Alencar, Donald D Cowan. “Artificial Intelligence versus Software Engineers: An Evidence-Based Assessment Focusing on Non-Functional Requirements”. June 2023. DOI:10.21203/rs.3.rs-3126005/v1.

Костюк В.С., Українець О.Р.
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Батько Ю.М, кафедра КІ ЗУНУ

ПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ТА ЇХ РОЛЬ В ОСВІТНЬОМУ ПРОЦЕСІ

Вступ. У сучасному освітньому світі підхід дистанційного навчання став невід'ємною частиною, особливо в умовах глобальних викликів та проблем, таких як пандемія COVID-19 та військові конфлікти. Платформи дистанційного навчання відіграють ключову роль у забезпеченні безперервності освіти, доступності навчальних матеріалів та інтерактивної взаємодії між учасниками освітнього процесу. Проте, веб-версії, що надають доступ до матеріалів даних платформ розроблені без врахування побажань кожного з користувачів і не дозволяють проводити гнучке налаштування. Це спричиняє деякий рівень дискомфорту при користуванні з ними. Тому створення та реалізація програмного додатку для доступу до курсів в системі Moodle є актуальною задачею [1].

Постановка задачі. Об’єкт дослідження – платформи для створення та підтримки систем дистанційного навчання. Предмет дослідження – відкрита платформа для дистанційного навчання Moodle. Головна мета даного дослідження полягає в дослідженні можливостей створення програмного додатку для доступу до платформ дистанційного навчання та виконання основних функцій роботи з ними. Розроблене програмне забезпечення дозволяє отримувати доступ до матеріалів курсів в системі Moodle, виконувати налаштування доступних функцій при роботі з платформою та здійснити налаштування інтерфейсу користувача в залежності від побажань користувача.

Основний матеріал. На сьогоднішній день існує велика кількість платформ, які надають можливість створювати онлайн курси різної складності, надавати до них доступ, організовувати діалог між викладачами та студентами. Серед основних, що представлені на сьогоднішній день є Google Classroom, Edmodo та Moodle.

Google Classroom – це безкоштовний веб-сервіс, розроблений компанією Google для навчальних закладів, що дозволяє організувати ефективний навчальний процес у віртуальному середовищі. Платформа дозволяє створювати віртуальні класи, організовувати завдання, оцінювання та зворотний зв'язок між викладачами та учнями. Інтеграція з іншими сервісами Google, такими як Google Drive, Google Docs, Google Meet, забезпечує зручний доступ до навчальних матеріалів та можливість проведення онлайн-уроків. Google Classroom особливо корисний у ситуаціях, коли необхідно швидко адаптувати навчальний процес до змінюваних умов, таких як карантинні обмеження або інші надзвичайні ситуації.

Edmodo – це освітня платформа, яка поєднує функції соціальних мереж з інструментами для організації навчального процесу. Вона дозволяє створювати віртуальні класи, публікувати навчальні матеріали, організовувати завдання та оцінювання, а також забезпечує можливість спілкування між учасниками освітнього процесу. Однією з особливостей Edmodo є наявність інтерактивних елементів, таких як вікторини, опитування та обговорення, що сприяють активному залученню студентів до навчання та розвитку їхніх комунікативних навичок. Платформа також забезпечує зручний інтерфейс для батьків, які можуть відслідковувати успішність своїх дітей та брати участь у навчальному процесі.

Moodle – це відкрита система управління навчанням, яка дозволяє створювати адаптовані онлайн-курси з різноманітними навчальними активностями, такими як тести, форуми, завдання, відео та інші ресурси. Викладачі мають можливість самостійно створювати курси, налаштовувати доступ до матеріалів, встановлювати часові обмеження, а також використовувати різноманітні інструменти оцінювання, що сприяє індивідуалізації навчання та розвитку самостійності студентів [2].

Moodle є розгалуженою системою управління навчанням, яка побудована на модульній архітектурі з відкритим програмним кодом. Платформа написана мовою програмування PHP та сумісна з різними типами баз даних, зокрема MySQL, PostgreSQL, MariaDB та іноді Oracle чи

Microsoft SQL Server. Ця гнучкість у виборі бази даних дозволяє адаптувати Moodle до конкретних технічних умов та інфраструктурних рішень навчального закладу або компанії. Moodle підтримує стандартні протоколи автентифікації користувачів, зокрема LDAP, OAuth 2.0, SAML, а також вбудовану реєстрацію. Це дає змогу інтегрувати його з внутрішніми інформаційними системами та забезпечити єдиний доступ (SSO) до навчального середовища. Система також підтримує SCORM, xAPI (Tin Can API) та IMS LTI, що дозволяє легко імпортувати курси з інших платформ або розгортати зовнішні навчальні інструменти без потреби в повному їхньому перенесенні. Інтерфейс Moodle базується на шаблонній системі з використанням HTML, CSS та JavaScript, що дозволяє налаштовувати зовнішній вигляд відповідно до брендингу навчального закладу. Завдяки темам (themes), які можна створювати або змінювати, Moodle легко адаптується до візуальних і навігаційних потреб конкретного користувача або організації. Крім того, він підтримує мультимовність і дозволяє додавати або перекладати будь-який інтерфейсний елемент без необхідності зміни ядра платформи.

Щодо архітектури, Moodle побудований на принципах MVC (Model-View-Controller) у дещо спрощеній формі. Він забезпечує API для розробки власних модулів, блоків, фільтрів і звітів. Завдяки цьому розробники можуть без суттєвого втручання в основний код реалізовувати нові функціональні можливості. Moodle також активно використовує систему подій (events) і тригерів (observers), що дозволяє логувати дії користувачів або виконувати додаткову логіку в момент настання певної події. З точки зору адміністрування, Moodle пропонує гнучку систему ролей та дозволів. Кожен користувач може мати кілька ролей у різних контекстах: наприклад, у межах конкретного курсу або категорії курсів. Це забезпечує точний контроль над доступом до ресурсів, завдань, тестів і системних функцій. Крім того, Moodle підтримує масштабування та високонавантажене середовище. Його можна розгорнути як на одному сервері, так і в кластеризованому варіанті з розподіленими компонентами – веб-сервером, сервером бази даних, файловим сховищем та кеш-сервером (наприклад, Redis або Memcached). Даний підхід значно підвищує можливості інтеграції даної платформи в навчальний процес навчальних закладів різного рівня [3].

Загалом Moodle є надзвичайно гнучкою, розширюваною та масштабованою платформою, яка завдяки своїй відкритості та спільноті розробників активно оновлюється та вдосконалюється, залишаючись одним із найпопулярніших рішень у сфері дистанційного навчання на глобальному рівні.

Висновки. Платформи дистанційного навчання, такі як Moodle, Google Classroom та Edmodo, суттєво змінюють традиційні підходи до організації освітнього процесу, забезпечуючи доступність, гнучкість та інтерактивність навчання. Вони сприяють розвитку цифрових компетентностей учасників освітнього процесу, підтримують індивідуалізацію навчання та забезпечують безперервність освіти в умовах змінюваного середовища. Однак успішне впровадження цих платформ вимагає залежить від наявних цифрових інструментів, що надають можливість доступу до функціоналу кожної з платформ. Використання сучасних технологій в платформі Moodle дозволяє розробляти програмні додатки, для роботи в даній системі, що значно підвищує рівень зручності під час користування нею.

Список літератури

1. Darren Turnbull, Ritesh Chugh, Jo Luck. “Learning Management Systems, An Overview”. *Encyclopedia of Education and Information Technologies*. 2020. pp.1052-1058. doi:10.1007/978-3-030-10576-1_248.

2. Бодненко Т. В. “Використання в LMS MOODLE у процесі навчання дисциплін з автоматизації виробництва майбутніх фахівців комп’ютерних систем”. *Наукові записки. Серія: Проблеми методики фізико-математичної і технологічної освіти*. Том 2, № 7, 2015. С. 21–26.

3. Славко Г. В. “Непрямі методи ідентифікації користувачів Moodle”. *MoodleMoot Ukraine 2020. Теорія і практика використання системи управління навчанням Moodle: тези доп. VIII міжнар. наук.-практ. онлайн конф. Київ*, 22 трав. 2020 р.

Моцал В.Р., Протаковський Р.А.
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Батько Ю.М, кафедра КІ ЗУНУ

ПРОГРАМНИЙ ЗАСІБ ТЕСТУВАННЯ ГЕНЕРАТИВНИХ МОДЕЛЕЙ ДЛЯ ВИЯВЛЕННЯ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ

Вступ. У сучасному інформаційному суспільстві зростає попит на програмні системи, здатні не лише ефективно генерувати контент на основі штучного інтелекту, а й здійснювати автоматизовану перевірку якості їхньої роботи. Зокрема, генеративні моделі, такі як нейромережеві системи для створення зображень, тексту або відео, вимагають розробки спеціалізованих підходів до тестування, оскільки традиційні методи верифікації не завжди є релевантними через стохастичну природу результатів. Одночасно з цим, завдання виявлення об’єктів, що рухаються у відеопотоці, має велике практичне значення для сфер відеоспостереження, автономного транспорту, робототехніки та розумних систем моніторингу. Тому створення та реалізація алгоритму створення запитів для виділення об’єктів у відеопотоці є актуальною задачею [1].

Постановка задачі. Об’єкт дослідження – технології обробки та розпізнавання даних у відеопотоці на основі технології штучного інтелекту. Предмет дослідження – моделі виділення об’єктів на основі словесного опису. Головна мета даного дослідження полягає в аналізу сучасних можливостей алгоритмів обробки природніх мов для створення коректних запитів для повного опису об’єктів у відеопотоці та алгоритмів пошуку вмісту за заданим описом. Отримані результати можуть бути використані для створення систем пошуку інформації в базах відеоданих на основі словесного запиту користувача.

Основний матеріал. Узагальнено, генератор тестових запитів можна визначити як автономну або вбудовану систему, що формує вхідні дані з урахуванням заданих параметрів і сценаріїв для подальшого використання в процесі тестування генеративної моделі. Основним завданням такого генератора є створення максимально репрезентативного набору запитів, здатного повною мірою охопити функціональні можливості досліджуваної системи та забезпечити якісну оцінку її результатів. Під час побудови генератора враховуються як семантичні характеристики вхідних запитів, так і статистичні аспекти їх розподілу в межах тестової вибірки [2].

Побудова генераторів може базуватися на декількох концептуальних підходах. Одним із них є генерація на основі граматичних правил або шаблонів, яка передбачає попереднє визначення структури запиту з фіксованими або варіативними компонентами. У цьому випадку можливим є чіткий контроль над синтаксичною правильністю запитів, проте обмеженою залишається їх семантична гнучкість. Альтернативою є використання стохастичних або евристичних моделей, зокрема марковських ланцюгів або трансформерних архітектур, які формують запити на основі ймовірнісного аналізу вхідних даних. У таких системах запити можуть мати високу варіативність, однак зростає потреба в додатковій перевірці їх релевантності. Також важливим є контекстно-орієнтований підхід, коли запити генеруються з урахуванням специфіки предметної області, наприклад, за наявності доменних знань про структуру відеоданих чи властивості цільових об’єктів [3].

Ключовим етапом процесу є не лише генерація самих запитів, а й визначення критеріїв, за якими буде оцінено якість відповідей моделі. У генеративних системах, де результат не є однозначним, важливо формувати обґрунтовану систему метрик, яка враховує як формальні, так і змістовні характеристики результатів. Серед найбільш поширених метрик виділяють ті, що вимірюють схожість результату з еталонними прикладами, наприклад, cosine similarity або BLEU у випадку текстових відповідей. Проте в багатьох випадках генерація є відкрита й не обмежена жорстким набором правильних відповідей, тому важливим є також використання оціночних показників, що відображають логічну узгодженість, цілісність, коректність та інформативність результату. У контексті комп’ютерного зору до критеріїв належать точність локалізації об’єкта, відповідність опису реальним візуальним характеристикам, а також узгодженість між

послідовними кадрами відео. Наприклад, якщо модель генерує опис руху об’єкта, то важливо визначити, чи відповідає згаданий напрям, швидкість або форма об’єкта тому, що зафіксовано у відео. У цьому випадку критично важливим є поєднання методів NLP та комп’ютерного зору з метою побудови узагальненої системи оцінювання, яка здатна працювати в умовах багатомодального середовища.

Приклад практичного застосування генератора тестових запитів можна розглянути на основі системи, що аналізує відеопотік і формує текстовий запит для пошуку об’єктів, які рухаються в кадрі. В такій системі генератор тестових запитів відповідає за створення вхідних сценаріїв або команд, які ініціюють запит до моделі на опис подій. Наприклад, один із запитів може бути сформульований як: *«Опишіть об'єкт, що рухається ліворуч у другій половині відео»*. Цей запит генерується з урахуванням часових обмежень, напрямку руху, а також умовного контексту, пов’язаного із сегментацією відео на часові інтервали. У відповідь модель повинна сформувати текст, наприклад: *«На 12-й секунді темна машина починає рухатись ліворуч повз будівлю»*. У цьому прикладі тестування буде базуватись на визначенні релевантності відповіді до запиту, а також перевірці відповідності між описом і візуальним контентом у відео. Для цього може використовуватись комбінація семантичного аналізу відповіді та автоматизованої обробки відеокадрів для визначення наявності відповідного об’єкта в заданому часовому діапазоні.

Особливу увагу необхідно приділяти якості тестових запитів. Вони мають бути не лише синтаксично коректними, але й лінгвістично різноманітними, щоб забезпечити стійкість моделі до варіативності мови. У разі використання природньої мови для взаємодії із системою, важливо враховувати синонімію, багатозначність та контекстуальну залежність формулювань. Це особливо критично для систем, які передбачають вільне формулювання запитів користувачем. Генератор повинен бути здатний створювати як прості, так і складні речення, включаючи уточнення, заперечення або порівняння. Крім того, у рамках побудови генераторів доцільним є використання напівавтоматизованих підходів, де частина запитів створюється вручну експертами, а далі використовується для тренування або налаштування генеративної моделі, що самостійно формує інші запити. Такий підхід дозволяє поєднати контрольованість з масштабованістю. Експерти забезпечують якість і релевантність, тоді як автоматичний компонент забезпечує обсяг і різноманітність.

Висновки. Генератори тестових запитів відіграють ключову роль у забезпеченні ефективної роботи генеративних моделей, особливо в умовах складних багатомодальних середовищ. Їх побудова вимагає комплексного підходу, що включає лінгвістичний аналіз, алгоритмічне проектування та доменну специфіку. Визначення критеріїв оцінювання відповідей моделі є не менш важливим етапом, що формує основу для об’єктивного аналізу її роботи. Приклад використання генератора у відеоаналізі демонструє практичну значущість таких систем у реальних застосуваннях, де потрібно описувати, класифікувати або інтерпретувати візуальні події. Очевидно, що розвиток таких підходів сприятиме підвищенню надійності та довіри до генеративних технологій в інтелектуальних програмних системах.

Список літератури

1. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. “Language models are few-shot learners”. *Adv. Neural Inf. Process. Syst.* 2020, *33*, pp. 1877–1901.

2. Jin, L.; Tan, F.; Jiang, S.; Köker, R. “Generative Adversarial Network Technologies and Applications in Computer Vision”. *Intell. Neurosci.* 2020, *2020*, 1459107.

3. Jabbar, A.; Li, X.; Omar, B. “A survey on generative adversarial networks: Variants, applications, and training”. *ACM Comput. Surv. (CSUR)* 2021, *54*, 157.

Assistant Professor Dr. Eng. Ihor Pazdriy
Department of Computer Science, Faculty of Engineering Sciences
Academy of Applied Sciences in Nowy Sącz

INTEGRATING GOOGLE ASSISTANT INTO THE SMART HOME CONTROL SYSTEM

Introduction. Initially, building management systems were used in private homes and apartments, but they were primarily developed for use in office buildings. Smart Home management systems were originally created by hobbyists; today, they are handled by professional companies [1].

Controlling lighting, multimedia, or heating is becoming a standard feature considered during renovation or construction. Nowadays, convenience and a sense of security are the top priorities. Moreover, more and more aspects of daily life are being automated, which allows people to save time and money, focus on important matters, and forget about certain duties. There is no need to worry — with a single press on a control device or a spoken command, all settings can be switched to predefined modes. It is also important that there are no limitations regarding specific locations or methods of communication. The best solution should enable connection to devices and command issuance from anywhere in the world. This technology continues to evolve — various switches, radio receivers, and remotes have been replaced by today's most popular solution: the Internet, which enables wireless communication between devices. This provides the ability to control them from anywhere in the world where Internet access is available via a smartphone or other network-connected devices.

Task Statement. To complete the research task, it is necessary to analyze the smart home management systems currently available on the market, particularly those that utilize voice commands. Research subject - Smart Home control systems using voice commands. Research object - Intelligent control systems for household appliances, in particular, a prototype of a Smart Home system based on Raspberry Pi and a mobile device.

The project involves building a prototype of a home management system using a Raspberry Pi microcontroller, a device running Android or iOS, and the required software. The next steps include analyzing the various possible applications of this prototype and conducting research to determine whether such a device is viable.

Primary Material. The Intelligent Building connects the installations within it in such a way that they communicate with each other and work together. This configuration makes it possible to create a system that is managed from one specific location such as a computer or panel. It is also important to bear in mind that these installations form unique subsystems consisting of a number of different devices, sensors and sensors [2]. Initially, only alarm, lighting and air conditioning systems were upgraded, but thanks to developments in computing and telecommunications, computer networks, automation and security systems have been introduced into the building enabling the monitoring and control of processes within it [2].

The pioneer of the application of voice control technology in Smart Homes is the FIBARO wireless system. In order to use it, the user needs to purchase the Fibaro Home Center 2/3 control panel or the Lite version. This central unit is indispensable. Without it, home automation devices would be mere objects. It collects and analyses information about the devices, connects them to each other, and thus controls the system's operation and ensures its reliability and safety. The calculations and operations of the system are performed locally via the control panel [3].

The Fibaro System and its devices may be controlled in a number of ways: through a mobile application provided by the manufacturer, using trigger devices such as remote controls or programmable buttons. Fibaro also allows for integrations with systems from other global brands, such as the Google Assistant voice assistant, Amazon's Alexa and Siri from Apple, which makes it possible to extend its functionality with the latest solutions [3].

Today, KNX Intelligent Home is the most popular home automation system in Europe and has been on the market for more than twenty years. The KNX standard brings together more than four hundred manufacturers including Merten, Berker, ABB, Gira, among others. The popularity of this system is growing all the time. The wide range of KNX devices allows for the most complex

realisations. Sensors, i.e. KNX pushbuttons and detectors, are powered by a safe 24V voltage, which is a great advantage of the KNX system. As a result, the buttons can be mounted close to the washbasin, shower cubicle or kitchen sink. The KNX Intelligent Home can be controlled via the Internet and telephone. Via these, in addition to visualisation, the user is able to change selected system parameters [4].

Based on the analysis, the systems were found to be complete and capable of managing the entire Smart Home. However, these systems are not perfect, because in the Fibaro wireless system, the user is exposed to connectivity problems, when there are a lot of devices, signal interference will appear. On the other hand, in the case of a system based on the KNX standard, a person who plans to implement such a facility in their home is practically deprived of this possibility, as it involves routing new cables in the walls. This solution is typically dedicated to new buildings in which the electrical installation is only just being planned [4].

The proposed simpler solution, focused only on the control of lighting whose 'brain' is based on the Raspberry Pi microcontroller. Using an Android phone using Google Assistant, the user utters a so-called trigger word, which is transmitted via the internet to the microcontroller, whereupon a specific GPIO port emits a signal to a switch and the light source lights up.

Google Assistant is a virtual assistant, an intelligent speech recognition service that is designed for mobile devices and smart home appliances. It is constantly supported by the developed machine learning algorithms and Artificial Intelligence, so that it successively gets to know its user and adapts to him or her, as its main task is to support the user in everyday life It is enough to ask him or her a question to receive the information needed at a given moment [5].

A prototype built using a Raspberry Pi 3B+ microcontroller to which a switch module was connected. The microcontroller was used as a device to control a relay that turns a light on or off. On the other hand, the device issuing the 'turn on', 'turn off' commands was a phone with Google Assistant installed, and the control was done using two separate applications, i.e. IFTTT and Particle, which had so-called actions implemented accordingly. By delving thoroughly into the idea of a Smart Building and understanding how other systems work, four hypotheses were made:

- the maximum distance from which the phone can be awakened to give commands to the Google Assistant is 3 metres,
- in the evening, the load on the relay is higher, resulting in slower internet speeds and therefore slower system response,
- the load on the microcontroller's processor and the high operating temperature have a significant impact on system performance,
- the device draws less current at rest than when it transmits a command to the relay.

The test of the maximum distance from which the phone can be woken up consisted of recording an audio file in which the Google Assistant's wake-up word is spoken, the wake-up word being 'Ok Google' by default. This recording then had to be played back from a loudspeaker for the results to be reliable, in this case a wireless loudspeaker was used with the volume set at a level comparable to how loud two people talk to each other. The speaker was then moved every 50 centimetres and the results recorded. The results of this test indicate that the Google Assistant can be put into a listening state from a distance of 2.5 metres. However, from a distance of 3 metres, the microphone was not able to pick up the sound for the Assistant to respond the first time. Also the hypothesis cannot be confirmed. The biggest influence on the test results was the listening device itself, the LG V30 phone on which the Assistant was installed. Newer devices with more microphones or greater sensitivity could recognise the command from a greater distance.

The relay load test consisted of performing internet speed tests using Ookla's Speedtest application and monitoring LTE signal parameters using the LTEWatch application, which has the ability to read them from the router (Huawei B525). Measurements were taken at four times of the day, i.e. morning (6:00), noon (12:00), evening (18:00) and night (24:00). The hours were chosen to show that the number of users accessing the Internet provided by the GSM transmitter at the same time has a significant impact on the Internet speed. The mobile base station, as measured using Google Maps, is 3.4km away from the router. The results of the survey indicate that Internet speeds are definitely lower during the evening hours, this is influenced by the higher load on the relay. It can be assumed that this is because the

majority of people working and studying are already at home at this hour and therefore using the Internet. Cable Internet is too slow for the majority of households, so they largely use LTE. In addition, during the course of the research, it was noticed that the speed of the Internet also affects the response time of the system to given commands, there is a slight delay between the transmission of information.

The microcontroller processor load test consisted of loading the processor to around 25%, 50%, 75% and 100%, and forcing RAM consumption to around 75%, the load was simulated using the stress-ng tool. This was followed by a check of the temperature at each load level. In addition, at a constant load level, measurements were taken of the microcontroller's response speed (5 measurements, from which the average response time was taken) to a given command to switch the light on or off.

The tests were carried out in such a way as to ignore the impact of Internet speed on the response time, i.e. the system was controlled without the Google Assistant and was controlled manually by issuing commands via the system terminal. From the data of this study, it can be concluded that a higher CPU and memory load does indeed have an impact on the speed of the microcontroller's response when performing a command such as turning on a light. The response time was 1.81 s, while with a load of around 100% it was 2.94 s. This means that the response time is 62% worse. However, the hypothesis that high temperature also affects the speed of the system must be rejected in this case. This is due to the fact that the device would have to reach an area of 70 degrees Celsius in order to start lowering the processor clocking, however, in the tests carried out it was not possible to reach a temperature higher than 65 degrees and no such event occurred.

The current draw test consisted of seven attempts to measure how much current the device draws in the idle state (IDLE), after issuing a command to start the light, after issuing a command to turn the light off. Measurements were taken using a current meter. The microcontroller was powered by the manufacturer's recommended charger with an output voltage of 5V and an output current of 2.5A. Measurements in the IDLE state were taken with the device with all cabling disconnected and Wi-Fi enabled. The measurements show that the hypothesis presented is true and the device draws less current in the idle state. The highest current consumption is when the microcontroller is maintaining the light on state, and on average 51mA lower when the light is off. This is due to the fact that when the light is on, the device needs to power the relay coil, which maintains a high state.

Conclusion. The experimental results confirmed the expected performance of the system. The first hypothesis was rejected, as the phone managed to wake up from 3 metres away, but the second time. Hypothesis two was confirmed, at 6pm the relay load was at its highest. Another observation was to note the slower response of the system to issuing commands, the lower internet speed resulted in slower processing of commands. The third hypothesis was also confirmed, that the load on the processor and the resulting higher component temperatures affected the speed of the system's response to commands. However, the differences were small. Between 0% and 100% load, the response time was 1.13s slower, which would not be noticeable in everyday use. The last hypothesis was also confirmed, the device was configured correctly and draws the least amount of current during idle and the most during operation i.e. switching on the light. Additionally, the current consumption per day of use was measured, which would be in the region of 48Wh.

References

1. Wikipedia, 2024, retrieved from: https://pl.wikipedia.org/wiki/Asystent_Google

2. Computer World, 2019. retrieved from: https://www.computerworld.pl/news/Asystent-Google-po-polsku, 411828.html

3. Fibaro, 2018 Downloaded from: https://www.fibaro.com/pl/products/home-center-2/

4. Control Home, 2017. retrieved from: http://controlhome.pl/system-knx/

5.Komputer Świat, 2018. retrieved from: https://www.komputerswiat.pl/artykuly/redakcyjne/Asystent_Google

Сеник Д.С.
Магістрант 2 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Шпінталь М. Я., кафедра КН ЗУНУ

ЕКСПЕРЕМЕНТАЛЬНІ ДОСЛІДЖЕННЯ АЛГОРИТМУ КЛАСИФІКАЦІЇ НА БАЗІ ТЕСТУ БЕЛЛА

Вступ. У сучасних задачах автоматизованої обробки природної мови однією з актуальних проблем є ефективне визначення тематичної близькості між текстами, особливо коли йдеться про великі обсяги неструктурованих даних, як-от енциклопедичні статті або інформаційні ресурси. Традиційні методи, зокрема ті, що базуються на векторних просторах, семантичних мережах або нейронних мовних моделях, часто вимагають значних обчислювальних ресурсів і попереднього навчання на великих корпусах. Це створює певні обмеження при їх застосуванні в умовах обмеженого доступу до апаратних потужностей або при роботі з низькоресурсними мовами, такими як українська. У цьому контексті актуальним є пошук простіших евристичних критеріїв, які дозволили б виявляти тематичну подібність між текстами без складних обчислень.

Одним із потенційно ефективних підходів є використання тесту Белла — математичного інструменту, що походить із квантової теорії і спочатку використовувався для оцінки кореляцій між квантовими системами. В останні роки цей тест адаптують для аналізу взаємозв'язків між лексичними структурами текстів, зокрема для виявлення "квантової заплутаності" між словами чи темами. Проте досі залишається відкритим питання щодо його ефективності в умовах реальних текстових даних, зокрема україномовних. Важливим аспектом також є дослідження, як саме параметри побудови HAL-матриць (зокрема розмір вікна) впливають на результати тесту Белла та чи справді отримані значення можуть бути інтерпретовані як маркер наявності спільної тематики у порівнюваних текстах. Наразі відсутнє систематичне вивчення цього підходу на корпусі текстів українською мовою, що ускладнює його інтеграцію в локальні інформаційно-пошукові системи. У зв'язку з цим виникає необхідність у формалізованому аналізі поведінки тесту Белла на прикладах із чітко визначеними тематичними межами.

Постановка задачі. Це дослідження має на меті перевірити застосовність тесту Белла як засобу для виявлення тематичної близькості між україномовними текстами, використовуючи для цього корпус енциклопедичних статей, тематично згрупованих у дві основні категорії — «домашні тварини» та «мови програмування». Основним завданням є оцінка того, чи здатен тест Белла фіксувати відмінності в семантичному контексті залежно від тематики тексту, а також аналіз того, як змінюється результат тесту при варіюванні розміру контекстного вікна у HAL-матриці. У межах цього підходу HAL-матриці використовуються як основа для побудови векторних представлень слів і документів, що дозволяє здійснювати порівняння між запитами й текстами без потреби у глибокому семантичному аналізі.

Основний матеріал. Ключовим аспектом є також встановлення, чи існує усталений поріг або характерна форма графіка залежності значень тесту Белла від параметрів моделі, яка б могла вказувати на релевантність документа до певної теми. Для цього досліджується, як поводяться значення тесту Белла при аналізі окремих текстів, що мають різний рівень узагальненості в межах обраної теми (наприклад, стаття «Домашні тварини» проти загальнішої статті «Тварини»). Очікується, що результати дозволять оцінити доцільність використання тесту Белла як інструменту для автоматизованої класифікації текстів за тематикою, зокрема в контексті опрацювання україномовного текстового корпусу без залучення великих обчислювальних ресурсів.

Варто зауважити, що високі значення тесту Белла не обов'язково свідчать про сильну заплутаність у квантовому сенсі, однак наявність тематичної відповідності тексту може корелювати з певним значенням розміру вікна при побудові HAL-матриці.

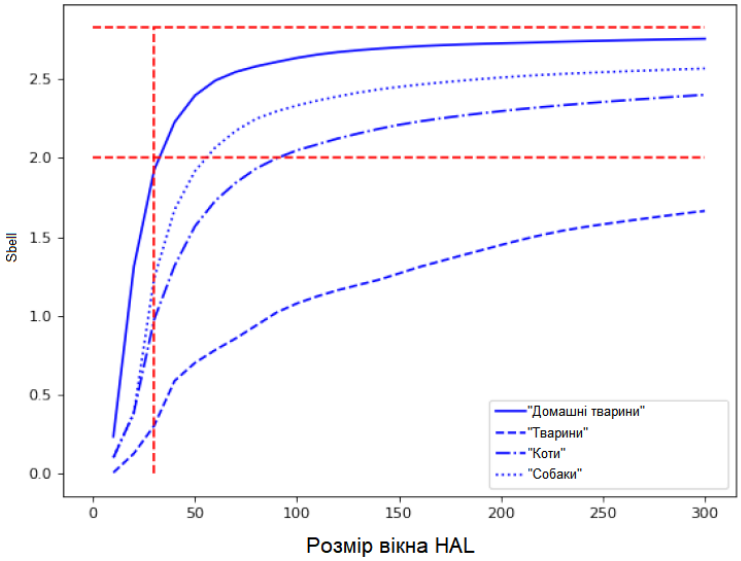


Рисунок 1 Результати розрахунку параметра Белла для тематики "Домашні тварини"

Аналіз значень параметра Белла для текстових документів засвідчив, що за певних умов — зокрема при варіюванні розміру контекстного вікна, що використовується для побудови HAL-матриці — можлива фіксація порушення нерівності Белла. Це свідчить про існування латентних взаємозв'язків у тексті, які не враховуються поточною моделлю. Такий підхід може розглядатися як інструмент для оцінювання тематичної близькості між текстовими документами. Крім того, для узагальнення поняття текстового запиту (який у дослідженні представлений лише двома словами) може бути використано аналогічну процедуру векторизації, що застосовується при формуванні векторів документів.

Висновки. У проведеному дослідженні було проаналізовано можливість використання тесту Белла як інструменту для виявлення тематичної близькості між україномовними текстовими документами. На основі корпусу енциклопедичних статей двох тематичних груп було показано, що результати тесту можуть відображати специфіку семантичного наповнення текстів залежно від обраної теми та ступеня її узагальненості. Варіювання параметрів побудови HAL-матриць, зокрема розміру контекстного вікна, дозволило зафіксувати випадки порушення нерівності Белла, що може слугувати індикатором наявності латентних тематичних зв'язків у текстах.

Отримані результати свідчать про перспективність застосування тесту Белла у задачах автоматизованої тематичної класифікації текстів без потреби в глибоких мовних моделях чи великих обчислювальних ресурсах. Особливої уваги заслуговує можливість адаптації цього підходу до україномовного корпусу, що є актуальним у контексті розробки інформаційно-пошукових систем для низькоресурсних мов. Подальші дослідження можуть бути спрямовані на розширення кількості тематичних груп, оптимізацію параметрів моделі та використання більш складних запитів для оцінки релевантності.

Список літератури

1. Кочан І. Лінгвістичний аналіз тексту [Текст]: навчальний посібник / І. Кочан. – К.: Знання, 2021. – 423.

2. A. Yu. Zubrytskyi, "Intellectual system of text research and analysis", M.S. thesis, National Technical University of Ukraine "Ihor Sikors'kyy Kyiv Polytechnic Institute, Kyiv, Ukraine, 2019 [in Ukrainian].

Довганюк К.А.[1], Зуброvич Р.Е.[2]
[1)]студент 4 курсу, [2)]магістрант ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Савка Н.Я., кафедра КІ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Вступ. В XXI столітті, а саме у сучасному цифровому середовищі, електронна пошта є основним каналом комунікації, зокрема, при реєстрації на різних сайтах, прив`язки пошти до акаунту в соціальних мережах, що робить її особливо вразливою до фішингових атак. У світі щодня надсилається велиезна кількість повідомлень, тому з метою безпеки, питання автоматизованої їх перевірки є ключовим. Електронна пошта є надзвичайно привабливим об`єктом для кіберзлочинців, які використовують фальшиві листи для введення в оману користувачів й, таким чином, отримання конфіденційної інформації: логінів, паролів, банківських реквізитів, тощо. Це, в свою чергу, несе негативні наслідки не тільки для окремих осіб, але й для цілих організацій.

На сьогодні існує чимало програмних інструментів, призначених для виявлення фішингових електронних листів, зокрема, SpamAssassin, Microsoft Defender для Office 365, Proofpoint Email Protection, Gmail Safe Browsing & Filtering. Основним їх недоліками виступають закритість природи алгоритму, складність адаптації й невисока достовірність ідентифікації фітингу. При цьому користувачі не можуть зрозуміти логіку прийняття рішень, налаштувати систему під власні потреби. [1].

Досить добре для задач класифікації зарекомендували себе методи машинного навчання. Їх алгоритми дозволяють розробляти моделі на основі реальних даних, враховувати складні закономірності у структурі та вмісті електронних листів, а також автоматично адаптуватися до нових загроз, що постійно з'являються.

Постановка задачі. Оскільки електронна пошта є особливо вразливою до оїфішингових атак з метою отримання конфіденційної інформації, існує потреба у розробці ефективного програмного засобу автоматизованого виявлення фішингових листів в електронних листах на основі методів машинного навчання із високою точністю.

Такий програмний інструмент повинен забезпечувати достовірне розпізнавання небезпечних повідомлень, ґрунтуватись на безпеці інформації, а також зменшенні ризиків фінансових чи репутаційних втрат в організаціях чи звичайних користувачів.

Об’єктом дослідження виступає процес виявлення фітингових атак.

Предметом дослідження – алгоритми та програмні засоби виявлення фітингових атак на основі методів машинного навчання.

Основний матеріал. Для реалізації програмного модуля перш за все необхідно провести попередній аналіз і очищення (будь-яких HTML-тегів чи видалення стоп-слів, тощо.) текстових даних, здійснити векторизацію вмісту у придатному для машинного навчання форматі. Наступним кроком є розробка моделей класифікації й у результаті їх порівняння оцінити точність, повноту та F1-метрику меделей. На основі оптимальної моделі класифікації провести реалізацію алгоритму виявлення фітингових атак й протестувати його на основі практичних задач.

Для роботи використано набір даних "Phish no more", який містить велику кількість розмічених прикладів електронних листів, які діляться на фішингові та легітимні. Такий датасет є одним з найпотужніших основ для навчання моделей, оскільки включає багато прикладів реальних фішингових посилань, що дозволяє моделі на його основі узагальнювати значення та виявляти закономірності. На першому кроці проведено попередню обробка, яка включає очищення тексту, нормалізацію, видалення стоп-слів та лематизацію. За перетворення тексту у векторну форму відповідають такі методи як, TF-IDF (термінова частота – зворотна частота документа) та CountVectorizer. Вони дозволяють представити текстові повідомлення у числовий вигляд, придатний для машинного навчання.

Саме результат векторизації впливає на точність класифікації. Для розробки моделей класифікації існують такі методи машинного навчання:

- Logistic Regression – базова лінійна модель, яка ефективна для бінарної класифікації;
- Naive Bayes – ефективний при роботі з текстовими даними;
- Random Forest – ансамблева модель, що об'єднує дерева рішень для досягнення високої точності;
- Support Vector Machine (SVM) – алгоритм класифікації для задач з чітко розділеними класами.

Під час моделювання протестовано декілька моделей машинного навчання з метою визначення їхньої ефективності для поставленої задачі. Дослідження моделей проведено на основі ключових показників, зокрема, повнота, метрики точності-якості та F1-міра, що є деяким середнім між точністю та повнотою. У таблиці 1 подано порівняльний аналіз моделей.

Таблиця 1 – Порівняння якості моделей машинного навчання

Модель	Точність	Повнота	F1-міра
Logistic Regression	0.95	0.94	0.945
Naive Bayes	0.92	0.89	0.905
Random Forest	0.96	0.95	0.955
SVM	0.94	0.93	0.935

Для побудови програмного модуля використано бібліотеки Python (scikit-learn, pandas, nltk, matplotlib.), які дають високу точність в практичних задачах та дозволяють ефективно класифікувати текст листів. Проте варто пам’ятати, що незважаючи на високу ефективність розглянутих моделей, існують певні обмеження, пов`язані із постійної еволюцією методів фітингових атак. Зважаючи на це, постає потреба в оновленні моделей із урахуванням нових типів атак [3].

Висновки.

Розроблено програмний компонент, який використовує підходи машинного навчання для розпізнавання фішингових листів. На основі зібраної бази даних фішингових та нефішингових листів виконано попередню обробку тексту, векторизацію даних, навчання та тестування існуючих моделей класифікації. Результати моделювання показали, що алгоритм “Random Forest” продемонстрував найвищу точність та стійкість класифікації.

Таким чином, розроблений модуль на основі машинного навчання є ефективним інструментом для автоматичного виявлення фішингу у вхідних електронних листах, мінімізує ризики виникнення помилок, що виникають через людський фактор, й сприяє підвищенню загального рівня кібербезпеки в цифровому просторі.

Список літератури

1. Abu-Nimeh S., Nappa D., Wang X., Nair S. "*A comparison of machine learning techniques for phishing detection.*" Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit. ACM, 2020. Pp. 60–69.
2. Basnet R., Sung A.H., Liu Q. "*Learning to detect phishing emails*." International Journal of Research in Engineering and Technology (IJRET), vol. 8, no. 3, 2021, Pp. 85–91.
3. Fang X., Zhang Y., Wang Y. "*Phishing email detection using improved machine learning techniques.*" Expert Systems with Applications, vol. 197, 2022. P. 116659.

Царук П.Р., Іванців Б.Я
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Батько Ю.М, кафедра КІ ЗУНУ

ПРОГРАМНА БІБЛІОТЕКА АЛГОРИТМІВ ОБРОБКИ ТА РОЗПІЗНАВАННЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

Вступ. Розвиток сучасних технологій зробив значний крок вперед за останні роки. Швидкі темки розробки нових апаратних засобів дали поштовх для створення нових більш складніших алгоритмів обробки даних, що вимагають більшу кількість операцій для прийняття кінцевого рішення. Якщо раніше використання таких алгоритмів потребувало значних часових затримок, то на сьогодні такі процеси можуть проходити в режимі реального часу та впроваджуватись в усе більшу кількість програмних систем. Тому створення та реалізація цифрової бібліотеки алгоритмів обробки цифрових зображень є актуальною задачею [1]

Постановка задачі. Об’єкт дослідження – алгоритми обробки та розпізнавання цифрових зображень. Предмет дослідження – цифрові зображення QR-кодів. Головна мета даного дослідження полягає в розробці структури та цифрової бібліотеки для отримання, обробки та розпізнавання цифрових зображень. Подальше використання даної бібліотеки дозволить спростити процес проектування та реалізації програмних систем з елементами обробки цифрових зображень.

Основний матеріал. Програмна бібліотека, призначена для обробки цифрових зображень, репрезентує собою інженерно-програмний комплекс, що забезпечує реалізацію широкого спектра операцій над візуальними даними, включаючи їх завантаження, трансформацію, фільтрацію, аналіз вмісту та автоматичну інтерпретацію. Основна функціональність таких бібліотек ґрунтується на використанні класичних методів цифрової обробки сигналів, математичного аналізу, теорії ймовірностей, статистики, а також сучасних підходів до машинного навчання, зокрема нейронних мереж.

Центральне місце в розробленій бібліотеці займають алгоритми попередньої обробки зображень, які дозволяють покращити якість даних шляхом усунення шумів, нормалізації яскравості, підвищення контрастності, а також зменшення розмірів і колірних перетворень. Ці етапи створюють передумови для подальшого ефективного аналізу зображень, зокрема для виділення структурних ознак, що є релевантними для задач класифікації, сегментації та розпізнавання.

Розроблена бібліотека побудована за модульним принципом, що передбачає розмежування функціональності на окремі логічні компоненти. Центральний модуль, відповідає за базову обробку зображень, тобто реалізує математичні операції згладжування, фільтрації, масштабування, геометричних трансформацій, бінаризації та морфологічних перетворень. На цьому рівні забезпечується стандартна підготовка зображень до більш високорівневої обробки.

Другий функціональний рівень включає модулі аналізу зображень, які використовують методи детектування контурів, обчислення градієнтів, виявлення ключових точок, а також алгоритми пошуку та класифікації об’єктів. Окремо слід відмітити алгоритми сегментації, що є основою для багатьох програмних систем обробки та розпізнавання цифрових зображень. Сегментація зображень є ключовим етапом у більшості систем комп’ютерного зору, оскільки вона забезпечує первинне структурне представлення зображення шляхом виділення значущих областей або об'єктів. Саме внаслідок сегментації система отримує можливість зосередитися не на всьому зображенні цілком, а на його семантично релевантних компонентах, що, в свою чергу, підвищує ефективність та точність подальших процедур розпізнавання, класифікації, вимірювання та інтерпретації.

В цифровій бібліотеці реалізовано декілька груп методів сегментації. Найпростішим методом є порогова сегментація, яка ґрунтується на аналізі інтенсивності пікселів [2]. У межах цього підходу кожному пікселю зображення приписується значення на основі порівняння його інтенсивності з певним граничним значенням. Якщо значення інтенсивності перевищує заданий поріг, піксель класифікується як такий, що належить до об’єкта, інакше — до фону. Також

реалізовані алгоритми сегментації на основі кластеризації, зокрема алгоритм k-середніх (k-means) [3]. Для даних алгоритмів сегментація трактується як задача розбиття простору ознак на k кластерів шляхом мінімізації внутрішньокластерної дисперсії. Кожен піксель зображення представляється в багатовимірному просторі ознак (наприклад, координати, яскравість, колір) і асоціюється з найближчим кластером. В бібліотеці реалізовані алгоритми на основі градієнтного аналізу. Вони передбачають виділення областей з високою інтенсивністю змін, які, як правило, відповідають межам між різними об'єктами на зображенні. Методи цього класу використовують оператори типу Собеля, Робертса або Кенні для виявлення меж. Ще одну групу складають алгоритми на основі графових моделей, серед яких варто відзначити методи нормалізованого розрізу (Normalized Cuts) та мінімального розрізу (Graph Cut). В таких алгоритмах зображення моделюється у вигляді неорієнтованого графа, де вершини відповідають пікселям, а ваги ребер — мірам подібності між сусідніми пікселями.

Наступним компонентом бібліотеки є програмний інтерфейс прикладного рівня, який забезпечує взаємодію користувача з основними функціями бібліотеки. До даного блоку належать функції завантаження/збереження зображень, організація під'єднання зовнішніх апаратних засобів для отримання цифрових зображень (сканер, фотоапарати) та виводу (принтер). Крім того, програмна бібліотека повинна невелику документацію та приклади застосування, що забезпечують більшу зручність в процесі її вивчення та використання під час проектування та розробки програмних систем обробки та розпізнавання цифрових зображень.

Як ілюстрацію практичного застосування бібліотеки розглянемо задачу автоматичного розпізнавання QR-кодів. На початковому етапі здійснюється захоплення зображень з камери з використанням відповідних інтерфейсів. Далі проводиться серія попередніх трансформацій, включаючи нормалізацію та фільтрацію шумів. На наступному етапі реалізується виявлення контуру qr-коду допомогою спеціалізованих алгоритмів виділення прямокутних областей. Завершальною фазою є обробка виділеної області цифрового зображення з виділений qr-кодом на основі правил розпізнавання кодів даного типу. В кінцевому результаті користувачеві буде продемонстровано веб-посилання, що було зашифровано в qr-коді.

Універсальність розробленої бібліотеки для обробки зображень дозволяє її широке впровадження в практику. Зокрема, в медицині може використовуються для аналізу результатів візуалізаційних досліджень біомедичних зображень, що дозволить підвищити точність діагностики патологічних процесів. У сфері безпеки технології розпізнавання обличь та об'єктів забезпечить ефективний процес моніторингу територій та виявлення загроз. В автомобільній галузі для проектування та реалізації систем автопілоту тощо.

Висновки. Розробка й впровадження програмних бібліотек алгоритмів обробки та розпізнавання цифрових зображень є ключовим етапом розвитку систем комп'ютерного зору. Вони забезпечують ефективне виконання широкого спектра задач, пов'язаних з інтерпретацією візуальної інформації, та є базовим компонентом інтелектуальних інформаційних систем нового покоління. Враховуючи стрімкий прогрес в галузі штучного інтелекту, зазначені бібліотеки стають дедалі потужнішими, відкриваючи перспективи для повної автоматизації процесів аналізу зображень у критично важливих сферах діяльності.

Список літератури

1. Лебедєва Олена, Нікова Діана, Трифонова Катерина. "Алгоритм сегментації цифрового зображення". *International Science Journal of Engineering & Agriculture. Vol. 2, No. 4, 2023*, p. 19-27. doi: 10.46299/j.isjea.20230204.03.

2. Грицик В.В. "Дослідження уніфікації стандартних порогових методів сегментації зображень". *Прикладні питання математичного моделювання*. Том 3 № 2.1 (2020): DOI: https://doi.org/10.32782/KNTU2618-0340/2020.3.2-1.

3. Oleh Berezsky, Oleh Pitsun, Grygory Melnyk, Vasyl Koval "Multi-threaded Parallelization of Automatic Immunohistochemical Image Segmentation". *Advances in Intelligent Systems, Computer Science and Digital Economics IV. CSDEIS 2022*. Lecture Notes on Data Engineering and Communications Technologies, vol 158. Springer, 2023 p. 266–275.

Горбатенко Д.О.
студентка 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Паздрій І.Р., кафедра КІ ЗУНУ

ПРОГРАМНІ ЗАСОБИ ДЛЯ ОБЛІКУ ТА ПЕРЕДАЧІ ПОКАЗНИКІВ ЕЛЕКТРОННИХ ЛІЧИЛЬНИКІВ ОБЛІКУ ЕНЕРГІЇ

Вступ. У зв'язку зі зростаючою потребою у точному та оперативному контролі за енергоспоживанням, значну актуальність набули системи автоматизованого обліку енергоресурсів. Традиційні засоби реєстрації показників електроспоживання, засновані на механічних або індукційних лічильниках, вже не відповідають сучасним вимогам до точності, оперативності та масштабованості обліку. Натомість на зміну їм приходять електронні лічильники, здатні не лише вимірювати спожиту енергію з високою точністю, а й передавати ці дані у реальному часі за допомогою вбудованих засобів зв'язку. З огляду на потребу централізованого збору, зберігання, аналізу та візуалізації енергетичних даних, постає завдання розробки й впровадження ефективного програмного забезпечення, здатного забезпечити комплексну взаємодію з електронними лічильниками. Такі системи мають вирішувати не лише функціональні задачі обліку, а й забезпечувати високий рівень надійності, захисту інформації та сумісності з різними протоколами та форматами передачі даних. Тому проведення класифікації технологій обліку та передачі даних є актуальною задачею [1].

Постановка задачі. Об'єкт дослідження – технології аналізу та передачі даних. Предмет дослідження – програмно-апаратні засоби обліку та передачі електроенергії. Головна мета даного дослідження полягає в систематизація підходів до реалізації програмних засобів для роботи з електронними лічильниками, аналіз існуючих рішень, а також формулювання основних вимог до таких засобів у контексті побудови ефективних інформаційно-керуючих систем енергомоніторингу. Отримані в процесі дослідження результати можуть бути використані для проектування та реалізації програмно-апаратних засобів автоматизованого обліку показників лічильника.

Основний матеріал. Сучасні електронні лічильники енергії здатні не лише акумулювати точні значення споживання в певні інтервали часу, але й зберігати ці дані у внутрішній пам'яті та передавати їх за запитом до централізованої системи. Така можливість породжує потребу у створенні складних програмно-апаратних систем, що оперують як локальними протоколами зв'язку, так і широкомасштабними мережами передачі даних. Ефективне функціонування таких систем передбачає не лише якісну технічну реалізацію з боку апаратної частини, а й гнучке та адаптивне програмне забезпечення [2].

У процесі розробки подібного програмного забезпечення важливим аспектом є забезпечення підтримки великої кількості протоколів зв'язку, що використовуються різними виробниками лічильників. Стандартизація у цій сфері залишається обмеженою, тому програмні продукти повинні враховувати різноманітність форматів, способів шифрування та архітектурних підходів до організації структури облікових даних. Основним завданням таких засобів є забезпечення цілісного зчитування інформації, її збереження в базі даних, захист від втрати та спотворення, а також можливість подальшої аналітики в реальному часі.

Ще одним критичним аспектом функціонування програмних систем є забезпечення безпеки інформації. З огляду на те, що показники споживання електроенергії можуть містити приватну інформацію про звички користувача, рівень захисту переданих даних повинен відповідати сучасним стандартам інформаційної безпеки. Це передбачає реалізацію механізмів шифрування, автентифікації, перевірки цілісності повідомлень та управління доступом до даних. У програмних реалізаціях активно застосовуються методи криптографічного захисту на основі сучасних симетричних та асиметричних алгоритмів.

Оскільки електронні лічильники, як правило, встановлюються на великій території, то однією з важливих функціональних вимог до програмних засобів є можливість централізованого або віддаленого доступу до даних. Для реалізації такої функціональності використовуються як традиційні канали зв'язку (наприклад, GSM, GPRS, 4G), так і новітні технології бездротового

обміну даними, зокрема LoRaWAN, NB-IoT та інші. При цьому програмна архітектура повинна бути адаптивною до умов нестабільного сигналу та здатною забезпечувати буферизацію даних у разі тимчасової втрати зв’язку [3].

Важливим фактором ефективності програмного забезпечення для роботи з лічильниками електроенергії є його масштабованість. У межах системи обліку можливе підключення як декількох десятків, так і кількох сотень тисяч приладів. Програмні компоненти повинні демонструвати високу продуктивність, а також гнучкість у налаштуваннях та оновленні, щоб не викликати збоїв у роботі всієї інфраструктури.

Окрема увага приділяється автоматизованим алгоритмам виявлення аномалій. Завдяки використанню засобів математичного аналізу, статистичних моделей та елементів машинного навчання можливо оперативно визначати підозрілі зміни у показниках енергоспоживання, що можуть свідчити про несанкціоноване втручання, технічні збої або інші відхилення від нормального функціонування. Таким чином, програмне забезпечення стає інструментом не лише фіксації спожитої енергії, а й забезпечення комплексного моніторингу її використання.

Розвиток програмних засобів у цій сфері також пов'язаний із тенденцією до інтеграції в екосистеми "розумного будинку" та "розумного міста". Передові розробки дозволяють інтегрувати дані з електронних лічильників у централізовані інформаційні системи управління інфраструктурою, що створює нові можливості для автоматизованого керування енергоспоживанням на рівні окремих домогосподарств, комунальних об’єктів та цілих мікрорайонів. У таких умовах програмне забезпечення має забезпечувати сумісність із різними платформами, відкритість API та підтримку взаємодії з іншими цифровими сервісами.

Трансформація енергетичної системи та перехід до цифрового енергетичного менеджменту вимагає постійного вдосконалення наявних рішень. Сучасні програмні засоби повинні не лише відповідати вимогам сьогодення, а й бути здатними до адаптації в умовах змінної технологічної реальності. Особливої актуальності це набуває у контексті впровадження енергетичних стратегій, орієнтованих на підвищення ефективності, зменшення втрат та забезпечення сталого розвитку енергетичної галузі.

У контексті сталого розвитку та декарбонізації економіки цифровізація обліку електроенергії виступає ключовим фактором забезпечення прозорості, підзвітності та раціонального використання енергоресурсів. Програмне забезпечення, що супроводжує процеси автоматичного збору та обробки показників, стає критичним елементом енергетичної інфраструктури. Його ефективність, безпека та надійність безпосередньо впливають на якість прийняття управлінських рішень та рівень енергетичної безпеки країни.

Висновки. Таким чином, програмні засоби для обліку та передачі показників електронних лічильників електроенергії відіграють стратегічну роль у процесах цифровізації енергетики. Їх розробка вимагає міждисциплінарного підходу, високого рівня стандартизації, врахування вимог надійності, безпеки та масштабованості. Від ефективності реалізації таких систем залежить не лише зручність споживача, а й ефективність функціонування енергетичної галузі загалом.

Список літератури

1. Wang M., Ren S., Du Y., Li Y., Ma X. Research on electric energy information acquisition system based on embedded platform. Multimedia Tools and Applications, 2024, Vol. 83, pp. 29377–29397.

2. Mustafa A., Cleemput S., Aly A., Abidin A. A Secure and Privacy-preserving Protocol for Smart Metering Operational Data Collection. arXiv preprint, 2018.

3. Liu Y., Chen Q., Hong T., Kang C. Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. arXiv preprint, 2018.

Бабюк Д.І.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ ВИЗНАЧЕННЯ ДІПФЕЙКІВ НА ЛЮДСЬКОМУ ОБЛИЧЧІ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ MEDIAPIPE

Вступ. В сучасному світі нейронні мережі розвиваються стрімкими темпами, це веде до більших можливостей, проте також й до більших загроз. Шахраї та зловмисники прагнуть виокристати нові інструменти на основі штучного інтелекту у своїх корисних цілях. Вони підроблюють голос, стиль письма, зовнішній вигляд людини, усе задля вигоди або інших мотивів, що порушують закон. Фальсифіковані зображення або відео носять назву “діпфейк”, від слова “deep” — глибокий, оскільки для їх створення використовуються моделі із поглибленним навчанням. Таким чином засоби, котрі можуть виявляти діпфейки з кожним днем стають все більш актуальний й насправді нагальним питанням.

Постановка задачі. Об’єкт дослідження — програмні засоби та моделі, що визначають діпфейки на людському обличчі. Предмет дослідження — використання Mediapipe у розрізі детекції обличчя та як один з елементів моделі для виявлення діпфейків. Мета дослідження — огляд наявних програмних засобів та моделей для виявлення діпфейків, створення програмного модулю, що зможе класифікувати діпфейки.

Основний матеріал. В ході дослідження було розглянуто три моделі глибинного навчання, до яких належать: згорткові нейронні мережі, генеративні моделі та рекурентні нейронні мережі. Розглянуто програмні засоби для виявлення діпфейків. Проведено аналіз роботи алгоритмів детекції обличчя та створено програмний модуль визначення діпфейків з використанням бібліотеки Mediapipe.

Методи поглибленого навчання зазвичай складаються з простіших моделей машинного навчання та застосовують послідовні операції для вилучення внутрішньої інформації з даних [1]. Машинне навчання (МН) визначається як дисципліна штучного інтелекту (ШІ), яка надає машинам можливість автоматично навчатися на основі даних і минулого досвіду, щоб виявляти закономірності і робити прогнози з мінімальним втручанням людини [2].

Згорткові нейронні мережі названі так через вхідні та вихідні шари, із якими вони працюють та над якими виконують операції “згортання”, тобто певну скалярну множину між матрицями зображення та фільтру, у результаті чого отримуються ознаки зображення. Такі мережі потребують великої кількості даних для навчання та вимагають процесори високої потужності (графічні або нейронні) задля достатньої швидкодії.

Генеративні моделі представляють собою такі моделі, що здатні на основі вхідних даних генерувати нові. Це стосується зображень, текстів, музики, відео. Автокодувальник є одним з типів генеративних моделей та працює за принципом стиснення, стискаючи та відновлюючи інформацію він виявляє ознаки та із використання справжніх зображень облич й вчиться. Коли буде використано несправжнє зображення під час відновлення воно буде виглядати гірше, ніж завантажене зображення й це можна використати для виявлення діпфейків.

Рекурентні нейронні мережі вирізняються своєю здатність обробляти послідовні дані, однак вони не мають змоги обробляти зображення напряму. Тому у вивленні діпфейків такі мережі використовують із CNN, що дістає ознаки із обличь та надає їх RNN, що аналізує їх й виявлеє динаміку руху чи певні аномалії, котрі свідчать що це діпфейк.

Програмні засоби для виявленні діпфейків представлені на більшості актуальних платформах та в тому числі у веб-просторі. Так Deepware Scanner достуний на Android та iOS й послуговується аналізом ознак стану обличчя після завантаження відео або посилання. Sensity AI є професійно системою, що працює через хмару та аналізує надані відео на глибокому рівні, враховуєьсбя фізика світла, синхронність губ і голосу, морфологія обличчя. Microsoft Video Authenticator подібна на Sensity AI через використання хмарної плафторми, цей сервіс найменші зміни у зображенні, фіксує мікропатерни: неузгодженості в тінях, спотворення текстур, надто “ідеальні” частини зображення, що характерні для результатів глибокого навчання. Hive

Moderation також використовує хмару для своїє роботи, сервіс застосовує розгалужену нейромережу, яка розпізнає не лише очевидні ознаки діпфейку, а й бібліотеку стилів діпфейкових генераторів, наприклад DeepFaceLab, FaceSwap або ZAO.

Однак перш ніж отримати результат чи перед нами діпфейк, чи справжня людина слід визначити чи перед нами взагалі об'єкт, що можна назвати людиною. Для цього використовуєтсья комп'ютерне бачення, куди входять алгоритми детекції, котрі здатні визначити що показано на відео чи зображенні. Комп'ютерний зір – це такі техногології, що дозволяють комп'ютеру отримувати та аналізувати дані із навколишнього серидовища, тексту, відео, звуку, тощо. В основі комп'ютерного зору лежить збір, обробка, аналіз та інтерпретація візуальних даних, отриманих за допомогою камер або сенсорів [3]. Серед алгоритмів детекції обличчя можна виділити класифіктаори на основі ознак Хаара та касккади локальни патернів.

Класифікатори на основі ознак Хаара являють собою фільтри у вигляді прямокутників (вертикальні, горизонтальні тощо) які мають дві сторони, одна світла (збільшення яскравості), інші темна (зменшення яскравості). В результаті накладання значення яскравності зі світлої та темної сторони спочаку додаються окремо, після віднімаються за принципом [чорна ознака] - [біла ознака], у результаті чого отримуємо ознаку (контраст), згідно якої можна класифікувати чи це обличчя. Каскади локальних бінарних патернів також використовують яскравість пікселів у алгоритмі роботи для визачення чи це обличчя. Центральний піксель порівнюється із кожним сусідом та записується значення 1, якщо [сусід ≥ центральний піксель], або 0 в інакшому разі. Згідно цього отримуємо 8-бітний бінарний код, котрий після перетворення буде ознакою. Після цього ознаки перевіряються каскадом класифікаторі й у разі їх класифікації можна говорити про те, що наприклад це обличчя або інший заданий об'єкт було виявлено.

Прогрмамний модуль використовує бібіотеки OpenCV для детекції обличчя й Mediapipe для виявлення ознак, у якості класифікатора обраний Random Forest. Для тестування роботи модуля був обраний датасет Deepfake Detection Challenge Dataset (DFDCD), що був створений для змагання Kaggle. У архіві мастітисься 10 Gb зразків зображень та json файл "matadta', що містить інформацію про справжні та фейкові зразки в архіві. Тож першочерговим завданням є розділення зображень після розпакування архіву на дві окремі папки, після потрібно дістати зображення з відео. Наступним кроком потрібно дістати зображення з відео. Фінальним етапом є запуск моделі та попереднє тренування. Вибрані архітектури нейронної мережі та навчені моделі зберігаються в репозиторії, що дозволяє проводити подальше навчання та експерименти. Остаточне рішення про класифікацію діпфейку приймається шляхом аналізу вихідних даних як модуля комп'ютерного зору, так і модуля обробки за допомогою машинного навчання, що забезпечує комплексну оцінку та кінцевий результат щодо автентичності відео. Результат роботи програмного модуля у вигляді матриці плутанини наведено на Рисунку 1.

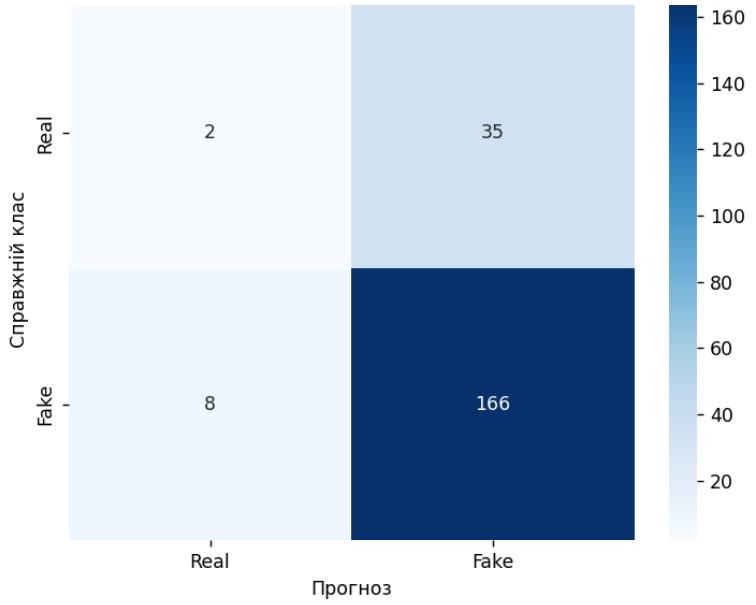


Рисунок 1 – Матриця плутанини

На даному етапі модель добре помічає фейкові зображення, проте через це переважна частина справжніх зображень класифікуються як фейкові помилково. Показник точності (~79.6%) близький до узагальненого значення для RandomForest (81 %), що й був використаний у моделі, він вищий за Logistic Regression (75 %), однак гірший за спеціалізовані модеілі чи CNN (85–88 %). За показником F1 для справжніх зображень результати є невтішними, 0.085 у порівнянні із Logistic Regression: (~0.45), CNN / XceptionNet-based (0.72–0.80), Random Forest: (~0.60). Однак у той же час F1 за виявленям діпфейків є навіть вищим за узагальнений для Random Forest, 0.885 проти 0.85, це все ще не 0.90+ як у CNN, однак результат на рівні.

Висновки. На основі аналітичного підходу проведено аналіз сучасних алгоритмів і програмних засобів виявлення діпфейків, що дозволило виділити їх переваги та недоліки і сформувати список вимог до програмного забезпечення. На основі підходів до розробки програмного забезпечення мовами високого рівня реалізовано програмний модуль для обробки відео та детекції ознак діпфейку на відео файлах довжиною до 10 секунд. Розроблено графічний веб-інтерфейс системи для підбору параметрів навчання нейронної мережі, що дозволило спростити процес навчання та реалізувати механізми для гнучкого налаштування параметрів.

Список літератури

1. A review of deep learning-based approaches for deepfake content detection / L. A. Passos et al. arXiv.org. URL: https://arxiv.org/abs/2202.06095

2. Kanade V. What is machine learning? Definition, types, applications, and trends. *www.spiceworks.com*. URL: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/

3. The Transmitted. Що таке комп'ютерний зір (Computer Vision, CV)? | TheTransmitted. TheTransmitted. URL: https://thetransmitted.com/adlucem/shho-take-kompyuternyj-zir-computer-vision-cv/

Петровський Ю.М.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ АНАЛІЗУ ВІДЕОПОТОКІВ З ВИКОРИСТАННЯМ LLM МЕРЕЖ

Вступ. У сучасному світі спостерігається стрімке зростання споживання медіа-контенту через Over-The-Top (OTT) платформи. Зростаюче навантаження на мережеву інфраструктуру, підвищення вимог до якості відеопотоків, а також необхідність забезпечення безперервного сервісу потребують ефективних засобів мережевого контролю та управління. У багатьох випадках медіа-оператори не мають точного уявлення про якість доставки контенту через різні мережеві протоколи та CDN інфраструктури, що призводить до погіршення користувацького досвіду [1].

Постановка задачі. Об'єкт дослідження – системи мережевого моніторингу OTT-потоків у реальному часі. Предмет дослідження – програмний засіб аналізу відеопотоків з інтелектуальними рекомендаціями на основі LLM технологій. Головна мета даного дослідження полягає в розробці системи мережевого моніторингу OTT-потоків, яка забезпечує збір, зберігання, обробку та візуалізацію даних про якість медіа-потоків в distributed network environments. Передбачається створення інтуїтивного мережевого інтерфейсу для операторів, який дозволить переглядати інформацію про network performance metrics, встановлювати мережеві пороги якості, а також отримувати автоматичні сповіщення у разі виявлення network-induced problems з інтелектуальними рекомендаціями щодо їх усунення.

Основний матеріал. У рамках розробки системи було здійснено детальний аналіз існуючих рішень у сфері мережевого моніторингу медіа-потоків та контролю якості OTT сервісів. Зокрема, були розглянуті такі системи, як Nagios для network monitoring, PRTG для bandwidth analysis та Zabbix для infrastructure monitoring. Найпоширенішими обмеженнями зазначених систем є відсутність специфічного аналізу OTT протоколів (HLS, RTSP, WebRTC), що унеможливлює точне відстеження якості медіа-потоків. Крім того, більшість платформ не підтримує інтеграцію штучного інтелекту для автоматичного аналізу мережевих проблем та генерування контекстуальних рекомендацій [2].

На етапі проєктування архітектури програмного забезпечення було створено діаграму компонентів мережевої системи (рисунок 1), що демонструє логіку взаємодії між основними network-oriented сутностями: CDN сервери, мережеві канали, network metrics, diagnostic reports, LLM integration для intelligent analysis тощо.

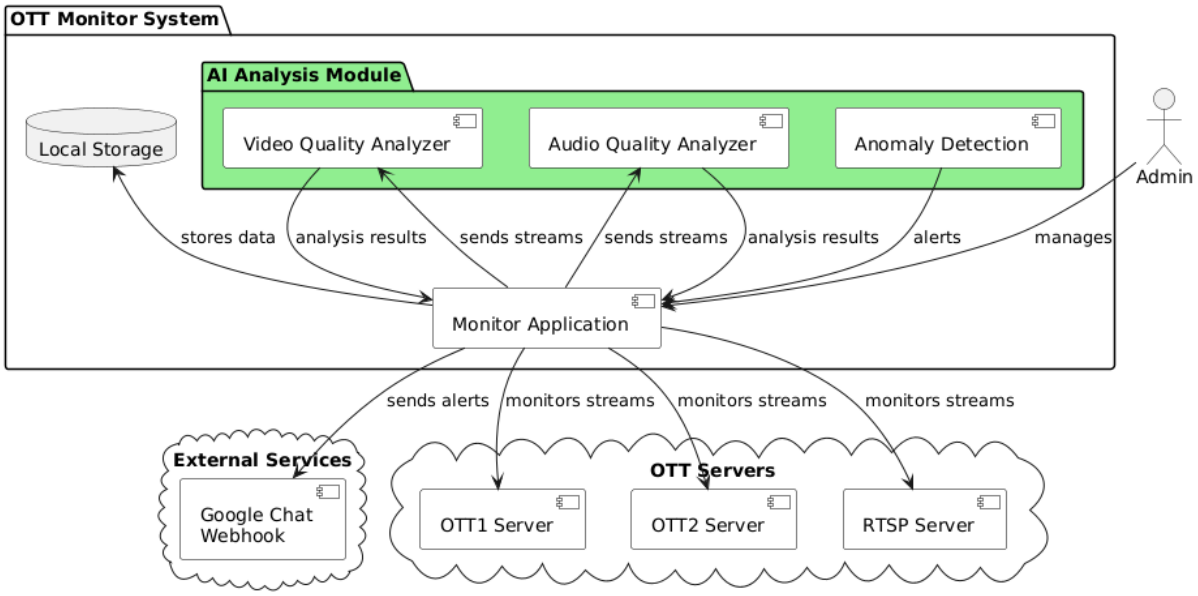


Рис. 1 – Діаграма компонентів мережевої системи

Для реалізації серверної частини обрано технологічний стек Python з asyncio для асинхронних мережевих операцій, OpenCV для обробки відеопотоків та tkinter для графічного інтерфейсу, що забезпечує високу продуктивність мережевих операцій. Всі компоненти розроблені у відповідності до мережевих стандартів з урахуванням network security та scalability. Клієнтська частина створена з використанням tkinter для desktop GUI з real-time network visualization capabilities та FFmpeg для аналізу медіа-потоків [3].

У таблиці 1 наведено порівняльний аналіз розробленої системи із аналогами.

Таблиця 1 - Порівняння розробленої системи з існуючими рішеннями

Система	OTT Protocol Analysis	Real-time Monitoring	LLM Integration	Bandwidth Analysis	Network Alerts	Video Preview
Розроблена система	+	+	+	+	+	+
Nagios	-	+	-	-	+	-
PRTG	-	+	-	+	+	-
Zabbix	-	+	-	-	+	-
SolarWinds	-	+	-	+	+	-

Розробка програмного забезпечення для моніторингу відеопотоків починається з написання модулів на мові програмування Python, здатних виконувати різні складні завдання, такі як аналіз мережевих протоколів, виявлення аномалій у bandwidth та генерування інтелектуальних рекомендацій. Цей процес включає в себе застосування численних бібліотек, наприклад, asyncio для асинхронного програмування та OpenCV для обробки відеопотоків. Система забезпечує автоматичне тестування якості мережевих з'єднань та моніторинг продуктивності OTT сервісів, що сприяє забезпеченню стабільності та ефективності процесу доставки медіа-контенту [4].

Висновки. Розроблена система мережевого моніторингу OTT-потоків дозволяє медіа-операторам ефективно контролювати якість доставки відеоконтенту через різні мережеві протоколи та CDN інфраструктури. Вона забезпечує гнучку інтеграцію з різноманітними OTT платформами, має зручний network-oriented інтерфейс, підтримує intelligent analytics з LLM integration та автоматичні мережеві сповіщення. Використання такої системи сприятиме більш ефективному управлінню мережевими ресурсами, підвищенню якості користувацького досвіду споживання медіа-контенту та зниженню operational costs для OTT провайдерів.

Список літератури

1. RFC 7826. Real Time Streaming Protocol Version 2.0 [Електронний ресурс]. – Режим доступу: https://tools.ietf.org/html/rfc7826
2. HTTP Live Streaming (HLS) Protocol Specification [Електронний ресурс]. – Режим доступу: https://tools.ietf.org/html/draft-pantos-hls-rfc8216bis
3. WebRTC API Documentation [Електронний ресурс]. – Режим доступу: https://webrtc.org/getting-started/overview
4. FFmpeg Documentation for Network Protocols [Електронний ресурс]. – Режим доступу: https://ffmpeg.org/ffmpeg-protocols.html

Чемерис М.М.
магістр, Національний Авіаційний Університет, м. Київ
Науковий керівник: д.т.н., професор Березький О.М., кафедра КІ ЗУНУ

АЛГОРИТМИ РОЗПІЗНАВАННЯ ЗВУКІВ ВИСОКОЇ ЧАСТОТИ

Вступ. У сучасному світі технічні засоби безпеки дедалі частіше включають аудіоаналітику як невід'ємну складову. Звукові події, що відбуваються у міському середовищі — постріли, вибухи, биття скла — часто стають передвісниками інцидентів. Їх своєчасне розпізнавання дозволяє скоротити час реагування відповідних служб. Особливу роль у цьому відіграє розпізнавання звуків високої частоти, адже вони є характерними для тривожних сигналів, сигналізацій, технічних аварій тощо. Засоби розпізнавання таких звуків передбачають створення ефективного програмного забезпечення, що здатне працювати в реальному часі з високою точністю [1] [2].

Постановка задачі. Об’єкт дослідження — процес розпізнавання звуків високої частоти. Предмет дослідження — алгоритми для виявлення і класифікації аудіо подій високочастотного спектра. Мета роботи — розробити програмний додаток для розпізнавання звуків високої частоти з подальшим реагуванням на ці сигнали у контексті безпеки.

Основний матеріал. Для навчання та тестування використовувався датасет UrbanSound8K, що містить понад 8700 урбаністичних аудіозаписів тривалістю до 4 секунд. Серед них: звук розбитого скла, сигналізація, сирени, постріли, гавкіт собак тощо. Записи розмічені за 10 класами, що робить його придатним для задач класифікації звукових подій [3]. У роботі були досліджені і використані алгоритми, які базуються на таких методах:

- DTW (динамічна трансформація часової шкали) – для базової подібності сигналів;
- HMM (приховані марковські моделі) – для моделювання звукових станів;
- SVM (метод опорних векторів) – для класифікації ознак звуків;
- GMM (модель гаусових сумішей) – для статистичного моделювання спектральних характеристик;
- Нейронні мережі – багатошаровий персептрон для глибокого розпізнавання з часового ряду.

Порівняння алгоритмів розпізнавання звуку наведено у таблиці 1.

Таблиця 1 – Порівняння алгоритмів розпізнавання звуку

Алгоритм	Точність розпізнавання (%)	Час класифікації (мс)
DTW	86	140
HMM	91	95
SVM	94	48
GMM	89	60
Нейронні мережі	96	65

Отже, за точністю розпізнавання найкращим алгоритмом є алгоритм, який побудований на основі SVM.

Оцінка точності розпізнавання звуків проведена на тестовій вибірці з датасету UrbanSound8K. В таблиці 2 наведено точність розпізнавання для різних типів подій.

Програмний додаток реалізовано на Python з використанням бібліотек librosa, scikit-learn, numpy [4]. Frontend частина створена на TypeScript. Система підтримує передачу попереджень у вигляді HTTP-запитів до API охоронних або сервісних служб.

Таблиця 2 – Точність класифікації за типами подій

Тип події	Точність (%)	Кількість перевірених зразків
Розбиття скла	92	55
Сигналізація	95	48
Постріл	94	53
Сирена	90	48
Ультразвук	88	50

Програмний додаток повинен виконувати такі функції:

- Мати здатність отримувати аудіо потік зі сторонніх носіїв таких як звуко записуючі пристрої та передавачі аудіо потоку.
- Працювати з різноманітними форматами аудіо.
- Вміти оброблювати аудіо доріжку по заданим параметрам та видавати результат аналізу.
- Мати можливість запускатися на різних пристроях незалежно від встановленої операційної системи.
- Надсилати результати обробки на виділені пристрої в режимі реального часу.
- Певний час зберігати аудіо матеріал та результати його обробки на виділених носіях.
- Надавати доступ до адміністрування параметрі та роботи з аудіо матеріалами.
- Користувач може обрати пристрій, аудіо доріжки з якого хоче послухати.
- Користувач може бачити на мапі місце, де встановлено звукозаписуючий пристрій.
- Користувач може бачити на мапі ділянку із підвищеною частотою.
- Розроблений додаток повинен бути кросс-браузерним веб-застосунком.
- Система повинна підтримувати до 100 користувачів одночасно.
- Система повинна відповідати на запит протягом однієї секунди.

Діаграма компонетів програмного додатку представлена на рисунку 2.

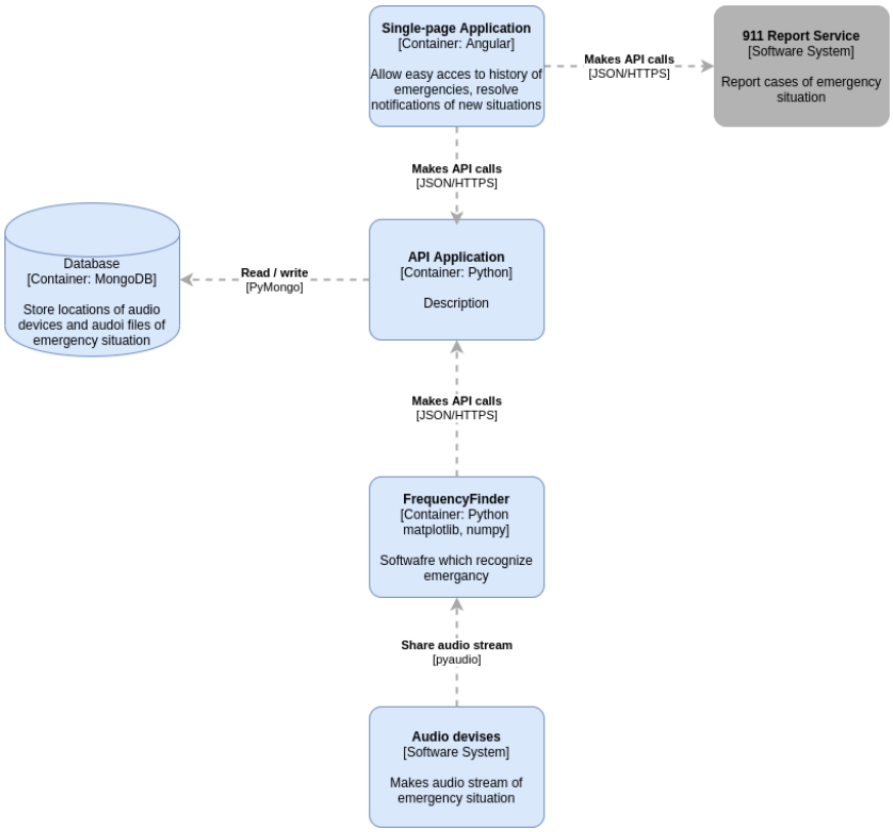


Рисунок 2 – Діаграма компонентів

Діаграма прецендентів представлена на рисунку 3.

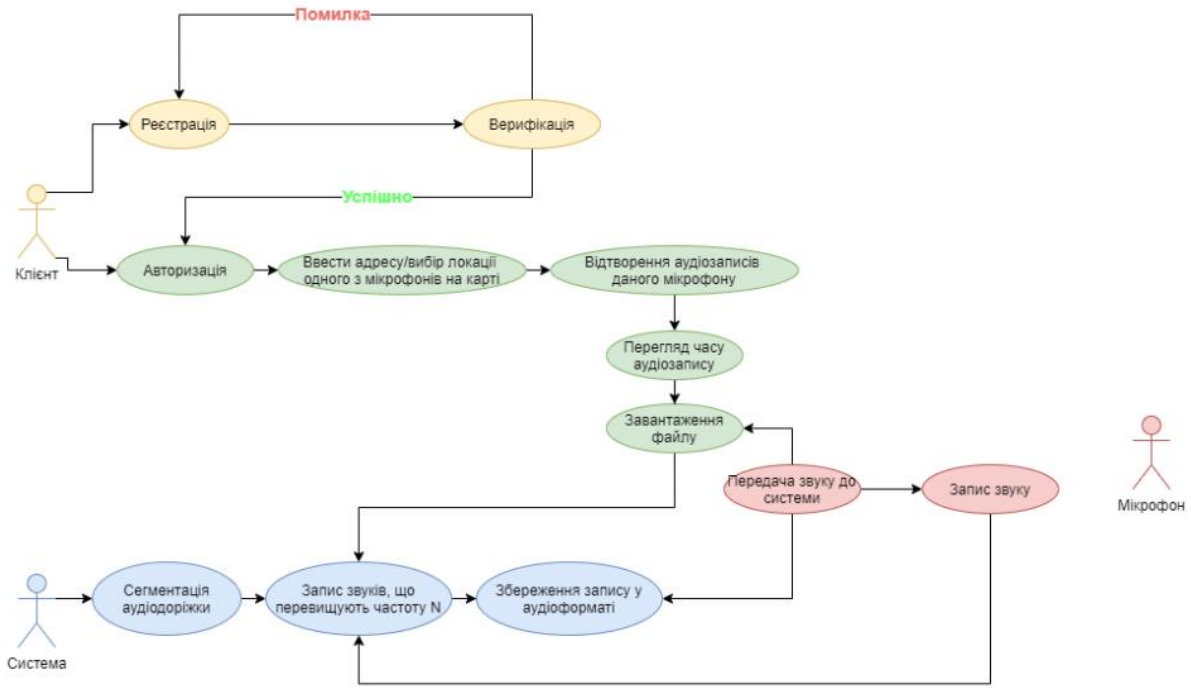


Рисунок 3. Діаграма прецендентів

Головний екран інтерфейсу представлений на рисунку 4.

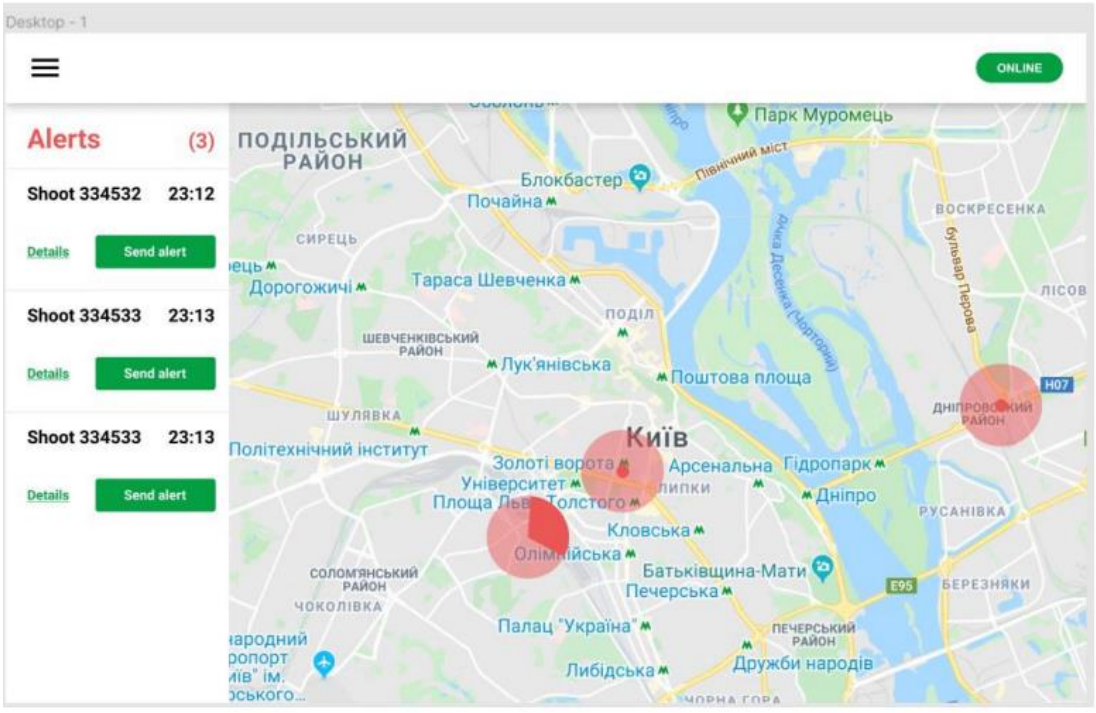


Рисунок 4 – Головний екран інтерфейсу з відображенням подій на карті.

Висновки. Проаналізовано п’ять основних алгоритмів розпізнавання звуків високої частоти. Розроблено ефективний програмний додаток розпізнавання звуків високої частоти, придатний для використання у системах безпеки. Перевагами додатку є висока точність, швидкодія, модульність реалізації та гнучкість у розгортанні. Програмний додаток може бути інтегрований у смарт-системи міської інфраструктури, охоронні комплекси приватних будинків, або як частина інтерфейсу доступності для осіб з порушенням слуху.

Список літератури
1. Rabiner, L., & Juang, B. (1993). Fundamentals of Speech Recognition. Prentice-Hall.
2. Jurafsky D., Martin J.H. Speech and Language Processing. Pearson, 2021.
3. ShotSpotter Inc. Technology Overview. URL: https://www.shotspotter.com/
4. Зибін С.В., Серебрякова С.В. Цифрова обробка сигналів у Python. НАУ, 2020.

Чемерис М.М.
магістр, Національний Авіаційний Університет, м. Київ
Науковий керівник: д.т.н., професор Березький О.М., кафедра КІ ЗУНУ

АЛГОРИТМИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ ПРОЦЕСІВ

Вступ. Логістичні процеси є критично важливими для сучасної економіки, особливо зараз, для нашої країни, оскільки забезпечують своєчасне переміщення товарів та ресурсів. Зі зростанням обсягів даних та потреби в автоматизації, зокрема в умовах глобалізації, оптимізація логістичних процесів стає ключовим викликом. Методи та алгоритми штучного інтелекту, особливо глибокого навчання та комп'ютерного зору, можуть суттєво підвищити точність та ефективність цих процесів. Проте, одна з основних проблем полягає у тому, що розпізнавання об'єктів на фото та відео в реальному часі досі залишається обмеженим через недостатню швидкість та точність алгоритмів. Оптимізація таких систем має важливе значення для зниження витрат, поліпшення обслуговування клієнтів і підвищення конкурентоспроможності компаній. Технології, що можуть ідентифікувати об'єкти на зображеннях із мінімальними помилками, дозволяють вирішити проблеми з відстеженням товарів, інвентаризацією та автоматичним управлінням запасами. У цьому контексті наукові дослідження з розвитку алгоритмів штучного інтелекту для логістики є актуальними [1,2].

Постановка задачі. Об’єкт дослідження – дослідження логістичних процесів, що потребують автоматизації шляхом розпізнавання об'єктів на зображеннях та відео. Предмет дослідження – алгоритми штучного інтелекту, що використовуються для розпізнавання об'єктів. Метою дослідження є порівняння алгоритмів штучного інтелекту для підвищення точності, швидкості та надійності розпізнавання об'єктів на фото та відео з метою оптимізації логістичних процесів. Для досягнення поставленої мети необхідно вирішити такі завдання:

- Проаналізувати існуючі методи і алгоритми штучного інтелекту для розпізнавання об'єктів (також оптимізатори, типи моделей, гіперпараметри тощо) [2].
- Визначити основні проблеми та обмеження, які перешкоджають їх ефективному використанню
- Розробити нові алгоритми або модифікувати існуючі для підвищення продуктивності та точності розпізнавання.
- Провести тестування розроблених алгоритмів на реальних даних
- Оцінити ефективність нових методів у контексті автоматизації логістичних процесів (за метриками Precision, Recall, F1, час виконання, кількість вхідних даних тощо)

Основний матеріал.

Проблема точності полягає у низьких рівнях розпізнавання об’єктів. Особливо це стосується конкретних випадків, коли потрібно розрізняти об’єкти, що подібні за більшістю, але відмінні за кількома параметрами (пляшки води різних літражів, упаковки товарів із ледве помітними відрізняючими факторами).

Проблема збору даних полягає у обмежених ресурсах набору і обробки датасету для навчання та валідації нейронних мереж

Проблема швидкості та ресурсів для прорахунку та отримання результатів у тій швидкості, що задовольнить потреби користувача. Для цього необхідні великі апаратні потужності, що є витратним. Дослідження базується на застосуванні математичних методів аналізу, моделювання та оптимізації алгоритмів штучного інтелекту. У роботі використані

методи глибокого навчання та нейронних мереж, а також методи комп'ютерного зору для розпізнавання об'єктів. Для оцінки ефективності було проведено серію експериментів з використанням відкритого датасету логістичних зображень SKU-110K. Моделі навчались із використанням GPU-інфраструктури. У дослідженні застосовуються методи глибокого навчання (CNN, YOLO, ResNet), а також гібридні підходи з класичним машинним навчанням. Реалізація базується на фреймворках TensorFlow, PyTorch та Azure Machine Learning. Особливу увагу приділено використанню трансферного навчання, що дозволяє адаптувати вже навчені моделі до нових наборів даних.

У дослідженні використані такі алгоритми, що використовувались при навчанні моделейнейронних мереж:

- YOLOv5 (базовий) – сучасний алгоритм одноетапного розпізнавання об’єктів, який оперує зображенням як єдиним блоком, прогнозуючи координати об'єктів і класи одночасно. Завдяки високій швидкості (FPS = 65) цей алгоритм підходить для реального часу.
- YOLOv5 з трансферним навчанням – адаптована версія, навчена на конкретному датасеті із попередньо натренованими вагами. Це дозволяє підвищити точність (до 91%) без повного перенавчання, особливо при роботі з обмеженою кількістю зображень.
- Faster R-CNN – двоетапна модель, яка спочатку генерує регіони-пропозиції, а потім класифікує об'єкти в них. Ця модель демонструє високу точність (Precision = 0.92), однак має нижчу швидкість (FPS = 24). Отже, ця модель тому більше підходить для офлайн-задач.
- SSD + MobileNet – легкий та швидкий алгоритм для мобільних пристроїв. Має високу продуктивність (FPS = 90), хоча поступається в точності іншим моделям [3]. Використаний датасет – SKU-110K, що містить понад 11 тисяч зображень магазинних полиць із більше ніж 1,7 мільйона вручну анотованих об'єктів. Він є орієнтованим на задачі логістики та складського розпізнавання [4].

Порівняння алгоритмів наведено у таблиці 1.

Таблиця 1 – Порівняння алгоритмів розпізнавання

Алгоритм	Precision	Recall	F1-score	Середня швидкість розпізнавання
YOLOv5 (базовий)	0.88	0.85	0.865	4,1с
YOLOv5 (transfer learning)	0.91	0.89	0.90	6,8с
Faster R-CNN	0.92	0.88	0.90	8с
SSD + MobileNet	0.83	0.81	0.82	3,3с

Висновки. Розроблені та протестовані алгоритми штучного інтелекту демонструють високу ефективність у вирішенні задач розпізнавання в логістиці. Зокрема, використання transfer learning дозволило зменшити потребу в великій кількості анотованих даних та суттєво прискорити навчання. Автоматизація логістичних процесів з використанням штучного інтелекту забезпечує скорочення часу обробки замовлень, зниження витрат на облік та інвентаризацію, підвищення точності обліку. Подальші дослідження зосередяться на оптимізації моделей для вбудованих пристроїв реального часу (Edge AI) [5].

Список літератури

1. Redmon, J., et al. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR.
2. Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition.
4. Goldman, E., et al. (2019). Precise detection in densely packed scenes. CVPR.
5. Chao, X., Hu, H., & Huang, K. (2020). AI in Logistics: A Review and Future Directions.

Ємець К.В.
аспірант 3 курсу кафедра СШІ НУЛП
Науковий керівник к.т.н., доцент Ізонін І.В., кафедра СШІ НУЛП

УДОСКОНАЛЕНИЙ МЕТОД ВІДБОРУ ЕКСПЕРТІВ У НЕЙРОМЕРЕЖЕВІЙ МОДЕЛІ СУМІШІ ЕКСПЕРТІВ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Вступ. Прогнозування часових рядів є критично важливим у фінансах, енергетиці, медицині та логістиці, де передбачення майбутніх тенденцій визначає ефективність планування та ухвалення рішень. Із розвитком глибинного навчання моделі на кшталт трансформерів та архітектур на основі механізму уваги продемонстрували високу результативність у роботі з послідовними даними завдяки здатності моделювати довгострокові взаємодії. Водночас такі підходи зазвичай потребують масштабних навчальних корпусів і краще працюють у багатовимірних постановках, що обмежує їхню ефективність для одновимірних рядів і задач з дефіцитом даних. Альтернативний напрямок пропонує парадигма «суміші експертів», первісно розроблена в контексті великих мовних моделей. Архітектури суміші експертів розширюють ємність моделі, підтримуючи низку спеціалізованих експертів і динамічно активуючи лише найбільш релевантну їх підмножину для конкретного входу. Такий підхід забезпечує високу інформативну здатність без надмірних обчислювальних витрат. У цій роботі подано модель прогнозування часових рядів, яка поєднує принципи суміші експертів [1] із новим механізмом відбору низки експертів, натхненним увагою. На відміну від класичних шарів відбору, що часто концентрують потоки на небагатьох експертах і вимагають додаткових балансувальних доданків у функції втрат, пропонований підхід використовує саму втрату прогнозування як сигнал для прийняття рішень маршрутизації. Метод відбору адаптивно переважує експертів відповідно до часових характеристик вхідних даних, даючи їм спеціалізацію на різних режимах прогнозування. Це дає змогу моделі можливість відтворювати як коротко, так і довгострокові часові залежності, уникаючи неефективності активації всіх експертів під час прогнозування. У підсумку цей метод є особливо дієвим для одновимірних рядів і сценаріїв з обмеженими даними, де багато наявних моделей, зазвичай, зазнають труднощів.

Постановка задачі. Об’єкт дослідження – процеси прогнозування часових рядів у умовах різних горизонтів прогнозування. Предмет дослідження – методи відбору експертів в нейромережевих моделях суміші експертів для часових рядів. Мета дослідження – удосконалити метод відбору експертів у нейромережевій моделі суміші експертів для забезпечення високої точності коротко- та довгострокового прогнозування часових рядів при зменшенні обчислювальних витрат, та емпірично підтвердити його переваги над аналогами.

Основний матеріал. Удосконалений метод відбору експертів перетворює кожне представлення часових значень на імовірнісний розподіл над множиною експертів [3]. Ключі утворюються афінною проєкцією часового ряду, а запити параметризуються спільними для всіх кроків навченими експертними векторами; для моделювання взаємодій вищого порядку між ключами й запитами застосовано добуток Кронекера замість стандартного скалярного добутку, що знижує колінеарність ознак. Отриманий тензор проектується у простір експертів, масштабується для числової стабільності та нормалізується нормована експоненційна функція. Внаслідок цього формується коректно нормований розподіл маршрутних ваг. Структуру всієї архітектури моделі для прогнозування часових рядів наведено на рисунку 1.

Для задач довгострокового прогнозування досліджувана модель із удосконаленим методом відбору експертів досягає найвищої загальної точності серед існуючих методів. Вона демонструє найменшу похибку на 4 наборах даних, зменшуючи середню абсолютну похибку приблизно на 4 % та на 8–9 % відносно найкращих базових рішень таких, як PatchTST і LSTM.

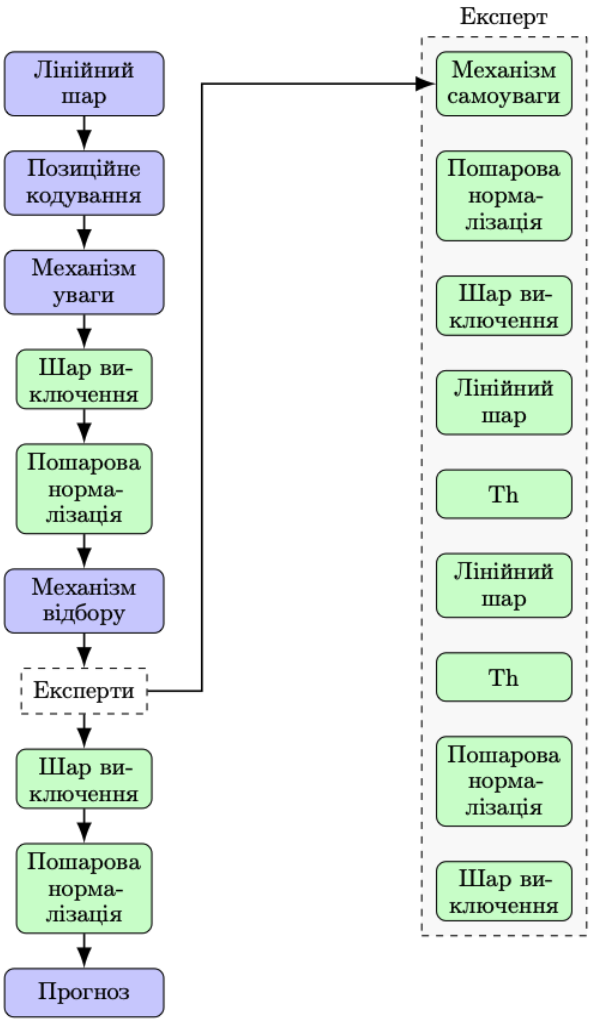


Рисунок 1 – Структура конфігураційного файлу

Важливо, що цих переваг досягнуто з меншим числом активних параметрів: менше, ніж у LSTM, і на 66% менше від PatchTST, що підкреслює вищу параметричну ефективність. Для короткострокового прогнозування досліджувана модель досягає найкращих абсолютних результатів, перевершуючи PatchTST приблизно на 2 % і випереджаючи LSTM на 5%.

Висновки. Удосконалено метод відбору експертів в нейромережевій моделі суміші експертів, який на відміну від наявного застосовує адаптований механізм самоуваги з параметризацією ваг через добуток Кронекера, що спрощує навчання моделі без використання додаткових балансувальних функцій втрат, збільшує різноманітність селективної маршрутизації експертів, та підвищує її точність при коротко- та довгостроковому прогнозуванні на 4–9 % відносно найкращих базових рішень.

Список літератури

1. Jacobs, R.A.; Jordan, M.I.; Nowlan, S.J.; Hinton, G.E. (1991). Adaptive mixtures of local experts. Neural Computation, 3(1), 79–87. https://doi.org/10.1162/neco.1991.3.1.79

2.Fedus, W.; Zoph, B.; Shazeer, N. (2022). Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, 23, 1–39. Available online: https://jmlr.org/papers/v23/21-0998.html (accessed on 25 April 2025).

3. Syntetos, A.A.; Boylan, J.E. (2005). The accuracy of intermittent demand estimates. International Journal of Forecasting, 21(2), 303–314. https://doi.org/10.1016/j.ijforecast.2004.07.009

Вовк Т. Ю.
Магістрант 2 курсу ІКНІ НУЛП
Науковий керівник к.т.н., доцент Ізонін І.В., кафедра СШІ, НУЛП

УДОСКОНАЛЕНИЙ АНСАМБЛЕВИЙ МЕТОД РЕГРЕСІЙНОГО АНАЛІЗУ МАЛИХ ВИБІРОК ДАНИХ

Вступ. У зв'язку зі стрімким розвитком технологій та зростанням інтересу до штучного інтелекту, методи машинного навчання дедалі ширше застосовуються для розв'язання задач регресійного аналізу у різних сферах. Досягнення високої точності таких підходів зазвичай потребує великих обсягів якісних даних. Проте формування подібних вибірок у різних прикладних областях є нетривіальним завданням. Складності зумовлені економічною недоцільністю або технічною неможливістю збору великої кількості спостережень на певний момент часу [1], тривалістю експериментів, а також етичними обмеженнями. За таких умов особливої актуальності набувають методи, здатні забезпечувати високу точність регресійного аналізу навіть при обмеженому обсязі даних.

Постановка задачі. Об'єкт дослідження - процеси інтелектуального аналізу коротких наборів табличних даних. Предмет дослідження – методи попереднього опрацювання та регресійного аналізу у випадку коротких вибірок даних. Мета дослідження полягає в підвищенні точності ансамблевого методу регресійного аналізу в умовах коротких наборів медичних даних за рахунок додаткового використання кластеризації.

Основний матеріал. Базовий метод регресійного аналізу – метод подвоєння входів, поданий у наукових публікаціях [2], [3], показав високу ефективність при роботі з невеликими наборами табличних даних, зокрема у задачах регресійного моделювання. Проте його застосування не завжди гарантує суттєве покращення. Це пояснюється обсягом та якістю вибірки даних, подекуди її складною геометричною структурую (нерівномірність кластерів, перекривання класів, тощо). З метою покращення результатів його роботи, у цьому дослідженні удосконалено цей метод, який передбачає використання кластерного аналізу для розширення вимірності простору вхідних даних задачі його результатами. Удосконалений метод має два режими: підготовки та застосування.

У режимі підготовки над початковим набором даних виконується кластеризація, в результаті чого всі об'єкти розподіляються по кластерах, а для кожного з них визначається центр та значення вихідної змінної (цільового атрибуту). Далі формується розширений набір даних, шляхом додавання до кожного вектора початкової вибірки нової ознаки - виходу центра кластеру $y_{\text{кластера\_}i}$ де $i = \underline{1, K}$ до якого цей вектор належить. Наступним кроком створюється аугментована вибірка даних, де вона розширюються по ширині та висоті, шляхом конкатенації усіх можливих двох пар векторів, а y_i - нове вихідне значення довільного подвоєного вектора (включаючи поточний) формується за допомогою формули $Z_{m,i} = y_m - y_i$ (де, y_m вихідне значення поточного вектора з опорної вибірки, y_i – вихідне значення для другого вектора з конкатенованої пари. Отримавши нову аугментовану вибірку, виконується процедура навчання нелінійного методу машинного навчання, яким у цьому випадку виступає SVR з радіально-базисним ядром.

У режимі застосування також формується розширений набір даних, шляхом додавання до кожного вектора тестової вибірки вихідної змінної центра кластеру $y_{\text{кластера\_}i}$, до якого належить поточний вектор. Для визначення приналежності вектора до певного кластера виконується порівняння Евклідової відстані до усіх центрів кластерів отриманих під час кластеризації навчальної вибірки, в результаті якого, вектор відноситься до того кластера, де ця відстань є найменшою. Наступним кроком почергово обирається вектор з розширеної тестової вибірки і формуються усі можливі пари з двох векторів, де на першому місці аугментованої вибірки буде значення поточного вектора тестової вибірки, а на другому місці усі вектори почергово з початкової вибірки. Після проведеної процедури аугментації застосовується метод машинного навчання натренований у режимі підготовки і формується остаточний результат із

використанням формули $y^{pred} \cong \frac{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} z_i^{pred}}{N}$ де, y_i значення з початкової опорної вибірки, z_i^{pred}– значення отримані після застосування методу машинного навчання, а N – розмір початкового набору.

Кластеризація передбачає попереднє визначення кількості кластерів і цей параметр має вирішальне значення для якості поділу вибірки [4]. Для пошуку оптимального значення параметру кількості кластерів було застосовано методи Silhouette Score та Prediction Strength. Оскільки отримані оцінки вказували на різні результати, подальші експерименти проводилися у межах від 2 до 7 кластерів.

Перевірка ефективності удосконаленого методу виконувалася на штучно згенерованому наборі даних сформованому за допомогою функції Франке який містить 100 векторів. Порівняння виконувалося як із базовим методом подвоєння входів, так і з класичною реалізацією SVR з радіально-базисним ядром, оскільки саме цей алгоритм використовується як у базовому, так і в удосконаленому варіантах для реалізації процедури навчання. Експериментальні дослідження показали, що використання удосконаленого методу з кількістю кластерів k=7, забезпечило зниження похибки RMSE майже на 40% відносно базового підходу, в той час як тривалість виконання помітно зменшилась (рисунок 1). Порівняно з класичною моделлю SVR точність прогнозування виявилася майже у 40 разів вищою, що підтверджує ефективність удосконаленого методу.

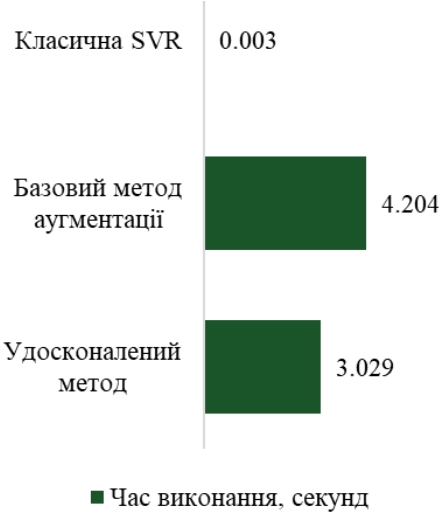


Рисунок 1 – Порівняння часу виконання кожного методу

Висновки. Дослідження показало, що удосконалений метод дозволяє досягти значного покращення показників ефективності порівняно з базовим методом. Отримані результати свідчать про практичну цінність розробленої модифікації та підкреслюють значимість використання результатів кластеризації у побудові моделей машинного навчання у випадку аналізу коротких наборів даних.

Список літератури

1. P. Xu, X. Ji, M. Li, and W. Lu, "Small data machine learning in materials science," 2023 *npj Computational Materials*, Abu Dhabi, United Kingdom, 2023, p. 42

2. Izonin, R. Tkachenko, S. Fedushko, D. Koziy, K. Zub, і O. Vovk, "RBF-Based Input Doubling Method for Small Medical Data Processing," Advances in Artificial Systems for Logistics Engineering, 2021, pp. 23-31

3. I. Izonin, R. Tkachenko, M. Gregus, K. Zub, і N. Lotoshynska, "Input Doubling Method based on SVR with RBF kernel in Clinical Practice: Focus on Small Data," Procedia Computer Science, 2021, pp. 606-613

4. E. S. Dalmaijer, C. L. Nord, and D. E. Astle, "Statistical power for cluster analysis," BMC Bioinformatics, 2022, p. 205