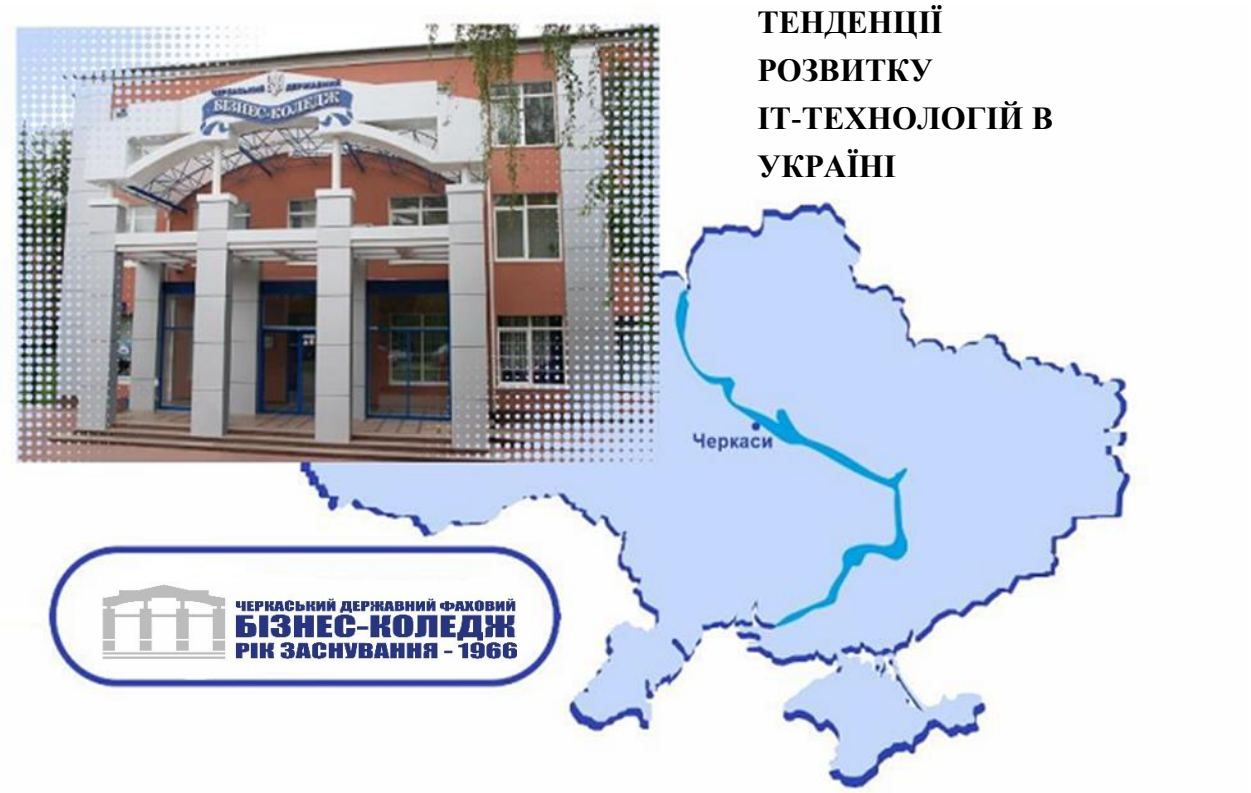


МАТЕРІАЛИ

XVIII Всеукраїнської

науково-практичної конференції

студентів, аспірантів та молодих вчених

за тематикою
«Тенденції розвитку
ІТ-технологій в Україні»

23-24 квітня 2026 р.
м. Черкаси

Міністерство освіти і науки України
Черкаський державний фаховий бізнес-коледж

МАТЕРІАЛИ

XVIII Всеукраїнської

науково-практичної конференції

студентів, аспірантів та молодих вчених

за тематикою

«Тенденції розвитку ІТ-технологій в Україні»

23-24 квітня 2026 р.
м. Черкаси

Матеріали XVIII Всеукраїнської науково-практичної конференції «Тенденції розвитку ІТ-технологій в Україні»: збірник наукових праць. Черкаси, 2026, 326 с.

Доповіді наукової конференції містять результати досліджень за наступними напрямками: штучний інтелект, обробка та захист інформації; інженерні підходи до розробки програмного забезпечення; інформаційні технології в галузевих рішеннях; робототехніка та адміністрування комп'ютерних систем.

Роботи друкуються в авторській редакції. В збірці максимально зменшено втручання в обсяг та структуру відібраних до друку матеріалів. Редакційна колегія не несе відповідальності за достовірність досліджень, матеріалів та результатів досліджень, що надано в рукописах, та залишає за собою право не поділяти погляди деяких авторів на ті чи інші питання, висвітлені в роботах.

Збірник становить інтерес для студентів, аспірантів, викладачів та наукових працівників.

Оргкомітет конференції

Азьмук Н.А. – голова оргкомітету доктор економічних наук, доцент, заступниця директора з навчально-методичної роботи Черкаського державного бізнес-коледжу.

Заболотній С.В. - професор кафедри інформаційних, мультимедійних технологій та дизайну ЧДБК, д-р тех. наук;

Хотунов В.І. – завідувач відділення бакалаврської підготовки ЧДБК, канд. пед. наук;

Захарова М.В. – завідувачка кафедри інформаційних, мультимедійних технологій та дизайну, доцентка кафедри інформаційних, мультимедійних технологій та дизайну, канд. тех. наук;

Бурмістров С.В. – доцент кафедри інформаційних, мультимедійних технологій та дизайну ЧДБК, канд. тех. наук;

Ночевнов Д.П. – доцент кафедри інформаційних, мультимедійних технологій та дизайну ЧДБК, канд. тех. наук;

Люта М.В. – завідувачка відділення інформаційних технологій ЧДБК;

Бреус Р.В. – відповідальний секретар, доцентка кафедри комп'ютерної інженерії та інформаційних технологій ЧДБК, канд. тех. наук.

Бреус Р.В. – відповідальний секретар, доцентка кафедри комп'ютерної інженерії та інформаційних технологій ЧДБК, канд. тех. наук.

ЗМІСТ

СЕКЦІЯ 1

ШТУЧНИЙ ІНТЕЛЕКТ, ОБРОБКА ТА ЗАХИСТ ІНФОРМАЦІЇ

УДК 004.056.5

ГІБРИДНІ ЗАГРОЗИ У СУЧАСНОМУ КІБЕРПРОСТОРІ

Янчишен Я. В.
yyanchishen@gmail.com
Черкаський державний фаховий бізнес-коледж
Захарова М.В.
м. Черкаси, Україна

Динамічний розвиток інформаційних технологій та глобалізація мережевого простору призвели до появи нових форм протистояння, де традиційні методи конфлікту замінюються гібридними впливами. Гібридні загрози у кіберпросторі представляють собою складний комплекс скоординованих дій, що поєднують технічні атаки на критичну інфраструктуру з інформаційно-психологічним тиском. Актуальність дослідження полягає у необхідності систематизації цих загроз для створення ефективних алгоритмів захисту та мінімізації ризиків у цифровому середовищі [1].

Мета роботи полягає у комплексному аналізі та систематизації сучасних гібридних загроз у кіберпросторі, дослідженні їхньої модульної архітектури та обґрунтуванні переходу до проактивної моделі захисту на основі інтелектуальних систем аналізу даних.

Одним із першочергових аспектів класифікації є розподіл загроз за механізмом їхнього впливу. Ключовим напрямком є техніко-деструктивні загрози, які спрямовані на порушення цілісності, доступності та конфіденційності даних. До цієї категорії належать цілеспрямовані атаки типу APT (Advanced Persistent Threat), використання вразливостей нульового дня (Zero-day) та експлуатація мережевих протоколів для несанкціонованого доступу до державних та корпоративних ресурсів [2]. Використання спеціалізованого шкідливого ПЗ (ransomware, wipers) стає інструментом не лише фінансового збагачення, а й стратегічного виснаження об’єкта атаки [3].

Важливим вектором гібридного впливу є інформаційно-когнітивні загрози. На відміну від технічних атак, вони спрямовані безпосередньо на користувача як «найслабшу ланку» системи безпеки. Ключовими елементами тут виступають

масові кампанії з дезінформації, використання бот-мереж у месенджерах (зокрема Telegram та Signal) для штучного формування громадської думки, а також соціальна інженерія. Сучасні алгоритми розповсюдження контенту дозволяють автоматизувати ці процеси, забезпечуючи високу швидкість та охоплення аудиторії при мінімальних витратах ресурсів [4].

Окрему увагу варто приділити інфраструктурно-ресурсним загрозам. Їхньою особливістю є комбінований вплив на автоматизовані системи керування технологічними процесами (SCADA). Атаки на енергетичні мережі, логістичні вузли та системи водопостачання створюють реальну загрозу фізичній безпеці населення. Важливим аспектом у цьому контексті є розробка механізмів виявлення аномалій у трафіку та впровадження принципів сегментації критичних мереж [5].

Логічно припустити, що архітектура сучасної гібридної атаки має модульний характер і може бути класифікована за функціональними компонентами (див.табл.1). Перспективним напрямком захисту є впровадження систем на основі машинного навчання (Machine Learning) для автоматичного аналізу логів та виявлення прихованих закономірностей у діях зловмисників. Це дозволить перейти від реактивної моделі безпеки («відповідь на інцидент») до проактивної, де потенційна загроза ідентифікується ще на етапі підготовки [7].

Оптимізація системи кібербезпеки в умовах гібридної війни вимагає не лише технічного переоснащення, а й розробки нових регуляторних стандартів. Важливим питанням залишається обмін даними про кіберинциденти між державним сектором та приватними компаніями. Це забезпечить створення централізованої бази знань про методи роботи хакерських угруповань та дозволить швидше реагувати на нові типи загроз [8].

Таблиця 1 – Функціональні модулі гібридної кібератаки

№	Назва модуля	Основна функція та інструментарій	Очікуваний результат
1	Проникнення та закріплення	Обхід Firewalls, IDS/IPS; використання Zero-day вразливостей.	Первинний доступ та стабільна присутність у системі.
2	Ексфільтрація даних	Приховане виведення конфіденційної інформації на зовнішні сервери.	Порушення конфіденційності; викрадення інтелектуальної власності.
3	Інформаційний супровід	Медійне висвітлення атаки, використання бот-мереж та дезінформації.	Штучне формування громадської думки, паніка.
4	Адаптивна протидія	Модифікація коду шкідливого ПЗ за допомогою штучного інтелекту.	Уникнення виявлення антивірусним ПЗ у реальному часі.

Проведене дослідження підтверджує, що сучасні гібридні загрози остаточно трансформувалися з суто технічних інцидентів у багатовекторні операції, де психологічний вплив на користувача є таким же критичним, як і атаки на SCADA-системи. Ключовим висновком є те, що ефективна оборона в умовах гібридної війни неможлива без впровадження Machine Learning для автоматичного аналізу аномалій, що дозволяє виявляти загрози ще на етапі їх підготовки.

У перспективі розвиток систем кіберзахисту має зосереджуватися на:

- Створенні інтелектуальних систем підтримки рішень, які допоможуть офіцерам безпеки діяти в умовах дефіциту часу.
- Поглибленні державно-приватного партнерства для оперативного обміну даними про нові методи роботи хакерських угруповань.
- Підготовці до нових викликів, таких як квантові обчислення та атаки на ланцюжки постачання ПЗ, що стануть визначальними у найближчі роки.

У майбутньому класифікація гібридних загроз може бути розширена шляхом додавання векторів, пов’язаних із квантовими обчисленнями та атаками на ланцюжки постачання програмного забезпечення. Подальші дослідження варто зосередити на створенні інтелектуальних систем підтримки прийняття

рішень для офіцерів кібербезпеки в умовах дефіциту часу та високої невизначеності.

Список використаних джерел:

1. Держспецзв’язку. Аналітичний звіт про кіберзагрози у світі та Україні. Державна служба спеціального зв'язку та захисту інформації України. URL: https://cip.gov.ua/ (дата звернення: 16.03.2026).
2. Горбулін В. П. Гібридна війна: всеохопний характер та інструменти реалізації. Національний інститут стратегічних досліджень. URL: https://niss.gov.ua/ (дата звернення: 16.03.2026).
3. ENISA Threat Landscape 2025. Reports on the state of cybersecurity in the EU. European Union Agency for Cybersecurity. URL: https://www.enisa.europa.eu/ (дата звернення: 16.03.2026).
4. EU DisinfoLab. Deepfakes and AI-driven disinformation: new challenges for security. EU DisinfoLab Publications. URL: https://www.disinfo.eu/ (дата звернення: 16.03.2026).
5. NIST. Guide to Operational Technology (OT) Security. National Institute of Standards and Technology. URL: https://csrc.nist.gov/ (дата звернення: 16.03.2026).
6. McKinsey Digital. The role of Generative AI in modern cyberattacks and defense strategies. McKinsey & Company. URL: https://www.mckinsey.com/ (дата звернення: 16.03.2026).
7. Microsoft Security Blog. Defending against hybrid threats: Machine Learning in cybersecurity. Microsoft. URL: https://www.microsoft.com/security/blog/ (дата звернення: 16.03.2026).
8. Закон України. Про основні засади забезпечення кібербезпеки України від 05.10.2017 № 2163-VIII (зі змінами станом на 2025 рік). Верховна Рада України. URL: https://zakon.rada.gov.ua/ (дата звернення: 16.03.2026).

УДК 004.056.53:004.8

ІНТЕЛЕКТУАЛЬНІ ПІДХОДИ ДО АВТОМАТИЗОВАНОГО ВИЯВЛЕННЯ ТА РЕАГУВАННЯ НА КІБЕРЗАГРОЗИ

Головко Д.І.
Paziloj444@gmail.com
Черкаський державний фаховий бізнес коледж
Бреус Р.В.
м.Черкаси, Україна

Сучасне цифрове середовище характеризується стрімким зростанням кількості та складності кіберзагроз – від класичних шкідливих програм до складних цілеспрямованих атак (APT). Організації змушені обробляти великі обсяги даних у режимі реального часу, що робить ручний аналіз неефективним. Традиційні сигнатурні системи захисту вже не забезпечують достатнього рівня безпеки, оскільки не здатні виявляти невідомі або модифіковані атаки.

У відповідь на ці виклики активно впроваджуються інтелектуальні підходи до кіберзахисту, які базуються на методах машинного та глибинного навчання, експертних системах і поведінковій аналітиці. Машинне навчання дозволяє моделювати поведінку загроз на основі історичних даних, зокрема класифікувати шкідливе програмне забезпечення та виявляти аномалії в мережевому трафіку. Глибинне навчання застосовується для аналізу складних структур даних, включаючи поведінкові патерни користувачів і взаємодії в мережі. Експертні системи забезпечують швидке реагування на відомі сценарії атак за допомогою правил і баз знань, а UEBA-технології дозволяють виявляти відхилення в поведінці користувачів і систем, що може свідчити про компрометацію облікових записів.

Підходи до виявлення загроз умовно поділяються на сигнатурні, аномалійні та гібридні. Сигнатурний аналіз ефективний лише для відомих загроз, тоді як аномалійний підхід дозволяє виявляти нові, раніше невідомі атаки. Гібридні методи поєднують обидва підходи для підвищення точності виявлення. Додатково використовуються методи аналізу мережевого трафіку та журналів

подій, що дозволяє ідентифікувати підозрілу активність, витоки даних або DDoS-атаки.

Важливим компонентом сучасних систем кіберзахисту є автоматизація реагування. SOAR-платформи координують роботу засобів безпеки та автоматизують сценарії реагування на інциденти. У разі виявлення загрози можуть автоматично блокуватися IP-адреси або облікові записи, ізолюватися заражені пристрої, генеруватися сповіщення для аналітиків, а також здійснюватися інтеграція з SIEM-системами для централізованого аналізу подій [3].

Інтелектуальні системи кібербезпеки мають низку переваг: вони забезпечують обробку великих обсягів даних у реальному часі, знижують залежність від людського фактору, здатні виявляти нові типи загроз, є адаптивними та масштабованими для великих інфраструктур. Це суттєво підвищує ефективність кіберзахисту [1].

Водночас існують і суттєві виклики. Для навчання моделей необхідні великі та якісні набори даних, спостерігається проблема хибнопозитивних спрацювань, а також складність інтеграції таких систем у вже існуючу інфраструктуру. Окремо слід виділити вразливість моделей до атак типу adversarial та етичні питання обробки даних [2].

Серед сучасних тенденцій розвитку кібербезпеки варто відзначити активне впровадження штучного інтелекту, перехід до архітектури Zero Trust, використання хмарних технологій, розвиток автономних систем реагування та інтеграцію безпеки у процеси розробки програмного забезпечення (DevSecOps) [4].

Практичне застосування інтелектуальних підходів охоплює корпоративні мережі, банківський сектор, державні інформаційні системи та IoT-середовища. Зокрема, вони використовуються для запобігання витокам даних, виявлення фінансового шахрайства та захисту критичної інфраструктури. Практика доводить, що без автоматизації процесів сучасний кіберзахист є недостатньо ефективним [3; 5].

Отже, інтелектуальні методи виявлення та реагування на кіберзагрози є ключовим напрямом розвитку сучасної кібербезпеки. Вони забезпечують швидкість, точність і масштабованість захисту, однак потребують подальшого вдосконалення, якісних даних та поєднання з експертною участю людини.

Список використаних джерел

1. Державна служба спеціального зв'язку та захисту інформації України. Офіційний вебсайт. URL: https://cip.gov.ua (дата звернення: 19.03.2026).
2. Національний координаційний центр кібербезпеки. Офіційний вебсайт. URL: https://ncscc.gov.ua (дата звернення: 19.03.2026).
3. CERT-UA. Офіційний вебсайт. URL: https://cert.gov.ua (дата звернення: 19.03.2026).
4. Закон України «Про основні засади забезпечення кібербезпеки України»: Закон України від 05.10.2017 № 2163-VIII. URL: https://zakon.rada.gov.ua/laws/show/2163-19 (дата звернення: 19.03.2026).
5. Національний банк України. Офіційний вебсайт. URL: https://bank.gov.ua (дата звернення: 19.03.2026).

УДК 004.8:004.93

ЕФЕКТИВНІСТЬ QLORA ДОНАВЧАННЯ МУЛЬТИМОДАЛЬНИХ МОДЕЛЕЙ В УМОВАХ ОБМЕЖЕНОГО ОБЧИСЛЮВАЛЬНОГО БЮДЖЕТУ

Цьома В. С.
tsyomav@gmail.com
Черкаський державний фаховий бізнес-коледж
Ночевнов Д.П.
м. Черкаси, Україна

Мультимодальні мовні моделі набувають широкого застосування в задачах розуміння зображень, однак їх використання у спеціалізованих предметних областях потребує додаткової адаптації. Запуск моделей у повному розмірі вимагає значних обчислювальних ресурсів, тоді як менші моделі можуть не досягати достатньої точності у вузьких задачах класифікації. Донавчання

дозволяє адаптувати попередньо навчену модель до конкретної задачі без необхідності навчання з нуля. Проте вибір методу донавчання, архітектури моделі та гіперпараметрів залишається нетривіальною задачею, особливо в умовах обмеженого обчислювального бюджету.

Метою цієї роботи є порівняльний аналіз двох мультимодальних мовних моделей – Gemma 3 4B [1] та Qwen 3.5 4B [2] – у задачі класифікації зображень за допомогою методу QLoRA в умовах обмеженого обчислювального бюджету. Як тестова предметна область використовувалися зображення покемонів першого покоління – набір із 151 класу.

Для дослідження було зібрано набір даних із зображеннями покемонів. Початковий набір містив зображення на білому фоні, з приблизно 45 зображеннями на клас. З метою підвищення узагальнювальної здатності моделей та імітації реальних умов використання було проведено розширення даних: до кожного зображення застосовувалися випадкові комбінації накладання шуму, зміни перспективи, горизонтального відображення та незначного повороту. Кінцевий набір даних налічував близько 20000 зображень, що відповідає приблизно 135 зображенням на клас. Для оцінки якості моделей було окремо зібрано тестовий набір даних із приблизно 2300 зображень шляхом скрапінгу з пошукової системи Bing. Тестовий набір не перетинався з навчальними даними.

Для порівняльного аналізу було обрано дві мультимодальні мовні моделі – Gemma 3 та Qwen 3.5, обидві розмірністю 4 мільярди параметрів. Вибір обумовлений обмеженим обчислювальним бюджетом дослідження – одним GPU NVIDIA A100 40GB та 100 compute units у середовищі Google Colab Pro, що дозволило провести повний цикл навчання обох моделей у межах наявних ресурсів.

Повне донавчання, хоча й забезпечує максимальну адаптацію моделі, потребує значних обчислювальних ресурсів і тому не розглядалося як основний метод у цій роботі.

LoRA є параметрично ефективним методом донавчання, за якого тренуються лише додаткові низькорангові адаптери [3]. Подальшим розвитком

цього методу є QLoRA, що поєднує 4-бітне квантування базової моделі з навчанням LoRA-адаптерів. Це дає змогу істотно зменшити використання пам’яті без критичної втрати якості [4]. Саме тому поєднання цих двох методів було обрано для донавчання.

Для стабілізації навчання при рангу r=32 разом із QLoRA використовувався підхід rsLoRA [5], який покращує масштабування внеску адаптера, зменшуючи нестабільність навчання.

Перед навчанням набір даних було нормалізовано до формату ShareGPT, що використовується фреймворком Unsloth для мультимодального навчання, де кожен запис містить системний промпт, зображення та очікувану відповідь. Для обох моделей використовувалася однакова конфігурація QLoRA: базові ваги квантувалися до 4-бітної точності (NF4), адаптери LoRA застосовувалися до всіх лінійних шарів моделі. Основні параметри наведено в табл. 1.

Таблиця 1 – Значення заданих параметрів адаптерів

№	Параметр	Значення
1	Ранг (r)	32
2	Alpha (α)	64
3	Dropout	0.05
4	rsLoRA	Увімкнено

Навчання проводилося протягом 3 епох з однаковими гіперпараметрами для обох моделей.

На рис. 1 наведено криві втрат обох моделей протягом навчання у логарифмічному масштабі.

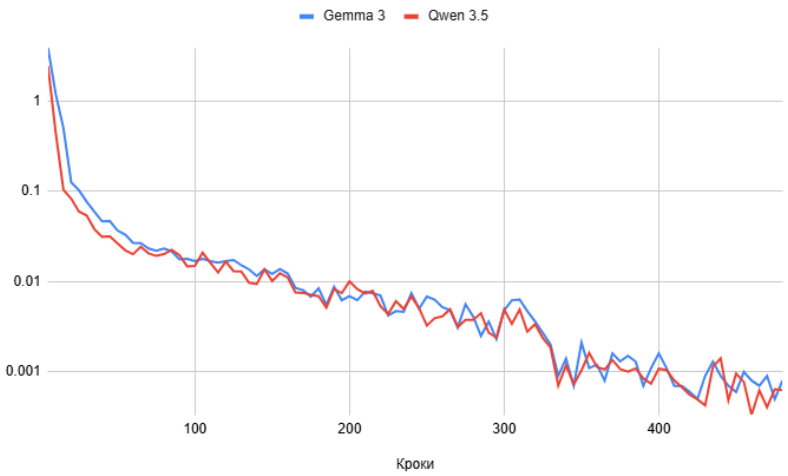


Рисунок 1 – Криві втрат Gemma 3 4B та Qwen 3.5 4B

Обидві моделі продемонстрували подібну динаміку зниження втрат і досягли близьких фінальних значень на навчальній вибірці. При цьому Qwen 3.5 показала дещо швидшу початкову адаптацію, що може свідчити про кращу придатність базової моделі до задач візуального розпізнавання.

Фінальним кроком стало тестування акуратності розпізнавання на попередньо підготованому датасеті. Результати наведено в табл. 2.

Таблиця 2 – Акуратність моделей на тестовому наборі даних

№	Модель	Епоха 1	Епоха 3
1	Gemma 3 4B	0.69%	82.2%
2	Qwen 3.5 4B	75.52%	80.92%

Отримані результати свідчать про швидшу початкову адаптацію Qwen 3.5 після першої епохи, тоді як Gemma 3, повільніше входячи в навчання, після трьох епох досягає вищої фінальної точності. Водночас Gemma 3 забезпечує швидший інференс, що робить її практичнішою за обмежених ресурсів.

Список використаних джерел:

1. Google Team. Gemma 3 technical report. 2025. URL: https://arxiv.org/abs/2503.19786 (дата звернення: 22.03.2026).
2. Qwen Team. Qwen3 technical report. 2025. URL: https://arxiv.org/abs/2505.09388 (дата звернення: 22.03.2026).
3. Hu E. J. та ін. LoRA: low-rank adaptation of large language models. 2021. URL: https://arxiv.org/abs/2106.09685 (дата звернення: 22.03.2026).
4. Dettmers T. та ін. QLoRA: efficient finetuning of quantized LLMs // Advances in Neural Information Processing Systems. 2023. URL: https://arxiv.org/abs/2305.14314 (дата звернення: 22.03.2026).
5. Kalajdzievski D. A rank stabilization scaling factor for fine-tuning with LoRA. 2023. URL: https://arxiv.org/abs/2312.03732 (дата звернення: 22.03.2026).

УДК 004.056.3:004.65

КОМПЛЕКСНІ МЕХАНІЗМИ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ ДАНИХ У СИСТЕМАХ РЕЗЕРВНОГО КОПІЮВАННЯ

Павлишин А. І.
arturpavlishin6@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р.В.
м. Черкаси, Україна

У сучасних умовах стрімкого розвитку інформаційних технологій та зростання обсягів даних питання їх надійного зберігання набуває критичної важливості. Інформація є одним із ключових ресурсів організацій і користувачів, а її втрата може призвести до фінансових збитків, порушення бізнес-процесів і зниження рівня довіри. Основними причинами втрати даних є апаратні та програмні збої, кібератаки, помилки користувачів і зовнішні впливи [1–3].

У зв’язку з цим особливого значення набувають методи забезпечення надійності даних, серед яких базову роль відіграють резервне копіювання та відновлення. Комплексний підхід, що поєднує резервування, реплікацію,

журналювання та контроль цілісності, дозволяє забезпечити високу доступність інформаційних систем і мінімізувати ризики втрати даних [1].

Основні методи забезпечення надійності даних включають backup, відновлення, реплікацію, журналювання та моніторинг систем. При цьому найбільш ефективним є їх комбіноване використання, оскільки кожен метод вирішує лише окрему частину задачі забезпечення відмовостійкості [4].

Особливу увагу приділено резервному копіюванню як базовому механізму захисту даних. Виділяють повне, інкрементне та диференціальне резервне копіювання. Повне копіювання забезпечує просте відновлення, але потребує значних ресурсів; інкрементне є більш економним, але ускладнює процес відновлення; диференціальне займає проміжне положення між ними [2; 3].

Процес відновлення даних є критично важливою складовою забезпечення безперервності роботи систем. Він включає відновлення з резервних копій та відновлення до визначеного моменту часу (Point-in-Time Recovery). Важливим етапом є регулярне тестування процедур відновлення, оскільки наявність копій не гарантує їх коректної працездатності [1].

Сучасні підходи також передбачають використання реплікації даних, кластеризації та хмарних технологій, що забезпечують високу відмовостійкість і доступність сервісів навіть у випадку інфраструктурних збоїв. Додатково важливу роль відіграють системи моніторингу, які дозволяють своєчасно виявляти аномалії та запобігати інцидентам [4].

Окремим аспектом є управління людським фактором. Впровадження політик розмежування доступу на основі принципу найменших привілеїв (PoLP), а також навчання персоналу з питань кібергігієни дозволяє значно знизити ризики випадкової або навмисної втрати даних.

Важливе значення має масштабованість рішень. Використання гібридних хмарних інфраструктур дозволяє поєднувати локальні ресурси з хмарними сервісами, забезпечуючи баланс між продуктивністю, доступністю та надійністю.

Додатково враховується нормативно-правова відповідність. Дотримання стандартів інформаційної безпеки, зокрема ISO/IEC 27000, та вимог законодавства дозволяє забезпечити юридичну стійкість і підвищити довіру до організації.

Таким чином, комплексний підхід до резервного копіювання та відновлення даних є ключовим елементом сучасних інформаційних систем, що забезпечує їхню відмовостійкість, безперервність роботи та захист критично важливої інформації.

Список використаних джерел:

1. De Novo. Backup (резервне копіювання). URL: https://denovo.ua/glossary/what-is-backup (дата звернення: 19.04.2026).
2. Artjoker. Резервне копіювання (backup). URL: https://artjoker.ua/big-brain/glossary/beckup/ (дата звернення: 10.04.2026).
3. HOSTiQ Wiki. Бекап – що це таке? URL: https://hostiq.ua/wiki/ukr/backup/ (дата звернення: 10.04.2026).
4. BDO Ukraine. Data backup and recovery services. URL: https://www.bdo.ua/uk-ua/services-2/bdo-digital/it-services/data-backup-and-recovery (дата звернення: 10.04.2026).

UDC 004.023

UTILIZING ARTIFICIAL INTELLIGENCE IN SOFTWARE DEVELOPMENT

Litvinov D.
daniillitvynov1@gmail.com
Cherkasy State Professional Business College
Prozorovska I. M.
Cherkasy, Ukraine

Today, artificial intelligence is an integral part of our lives. It is used for everything from providing helpful advice for daily routines to developing complex software products.

Let's explore how artificial intelligence can be utilized in learning programming and its subsequent application as a powerful assistant for professional developers.

There are many different LLMs (Large Language Models) available today. Currently, Claude is considered the most powerful and popular choice among programmers for writing code.First, let's define what an LLM is: A large language model (LLM) is a type of artificial intelligence (AI) program that can recognize and generate text, among other tasks. LLMs are trained on huge sets of data – hence the name «large». LLMs are built on machine learning: specifically, a type of neural network called a transformer model [1].

While models like ChatGPT and Gemini are widely used, Claude is often the preferred choice for programming tasks. What is Claude? Claude is a family of proprietary large language models (LLMs), as well an AI assistant and other AI tools powered by those models, developed by Anthropic. Claude models, particularly from their third generation onward, have consistently ranked among the top performing generative AI models available on the market [2].

Research across various models has shown that Claude often delivers the most effective results. For students learning to code, Claude can be used to:

- Create structured study plans.
- Master the fundamentals of programming.
- Explore various libraries and frameworks.

Similarly, professional programmers integrate Claude into their workflow to enhance their own skills. This synergy optimizes work, increases efficiency, and minimizes the risk of critical errors. To achieve the best results, a programmer must learn how to formulate task requirements correctly. In the world of AI, this is known as a prompt, or more specifically for development, Prompt Programming. It is a new paradigm in software development, where programs are created through the use of text-based instructions, known as prompts.

Instead of writing code in a programming language, you describe to the model in natural language what you want it to do. The model then generates the corresponding code.[3]. The more precise the prompt you create, the higher the quality of the resulting code. Furthermore, Claude can independently review code written by a programmer, identify errors, and suggest improvements.

Artificial intelligence is constantly evolving. To remain an effective professional in the programming field, you must continuously acquire up-to-date skills to minimize development time. Therefore, mastering prompting is one of the most vital tasks in our modern reality.

References:

1. What is a large language model (LLM)? URL: https://www.cloudflare.com/learning/ai/what-is-large-language-model/#:~:text=A%20large%20language%20model%20(LLM)%20is%20a%20type%20of%20artificial,network%20called%20a%20transformer%20model (дата звернення: 09.04.2026).
2. What is Claude AI? URL: https://www.ibm.com/think/topics/claude-ai (дата звернення: 09.04.2026).
3. Fundamentals of Prompt Programming. URL: https://www.authorea.com/users/766114/articles/918446-fundamentals-of-prompt-programming-introduction (дата звернення: 09.04.2026).

УДК 004.056.53

МЕТОДИ ВИЯВЛЕННЯ КІБЕРЗАГРОЗ У КОМП'ЮТЕРНИХ МЕРЕЖАХ

Хлівенко Р.А.
r.hlivenko@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р.В.
м. Черкаси, Україна

Зі зростанням масштабів використання комп'ютерних мереж та цифрових сервісів значно підвищується кількість і складність кіберзагроз. Сучасні атаки характеризуються високим рівнем автоматизації, адаптивністю та здатністю обходити традиційні засоби захисту. У зв'язку з цим методи виявлення кіберзагроз стають критично важливим компонентом систем інформаційної безпеки. Ефективність цих методів визначає здатність своєчасно ідентифікувати потенційні атаки, мінімізувати їх наслідки та забезпечити безперервність функціонування інформаційних систем.

Основною метою даного дослідження було проаналізувати основні методи виявлення кіберзагроз у комп'ютерних мережах, дослідити їх особливості реалізації та визначити ключові переваги й обмеження з точки зору ефективності, точності та продуктивності.

Методи виявлення кіберзагроз у комп'ютерних мережах поділяються на кілька основних класів, серед яких найбільш поширеними є сигнатурні, аномалійні та гібридні підходи [1].

Сигнатурний підхід базується на використанні заздалегідь визначених шаблонів атак (сигнатур). Кращі результати цей метод демонструє при виявленні відомих загроз, оскільки дозволяє швидко ідентифікувати характерні ознаки шкідливої активності. Проте його ефективність суттєво знижується у випадку нових або модифікованих атак, для яких сигнатури ще не сформовані [2].

Аномалійний підхід орієнтований на аналіз поведінки мережі та виявлення відхилень від нормального стану. Фактично це означає, що система формує модель «нормальної» активності, після чого всі нетипові дії розглядаються як потенційні загрози. Кращі результати досягаються при використанні

статистичних методів і алгоритмів машинного навчання, що дозволяє виявляти раніше невідомі атаки. Водночас цей підхід характеризується підвищеною кількістю хибних спрацювань [3].

Гібридні методи поєднують сигнатурний та аномалійний підходи, що дозволяє компенсувати їхні недоліки. У таких системах одночасно використовується база відомих атак і механізми аналізу поведінки, що підвищує загальну ефективність виявлення загроз.

Порівняльна характеристика основних методів виявлення кіберзагроз наведена в табл. 1.

Таблиця 1 – Порівняльна характеристика методів виявлення кіберзагроз

№	Функціональний компонент	Сигнатурний метод	Аномалійний метод	Гібридний метод
1	Принцип роботи	Пошук відомих сигнатур	Виявлення відхилень від норми	Комбінування підходів
2	Ефективність	Висока для відомих атак	Висока для нових атак	Висока загальна
3	Хибні спрацювання	Низькі	Високі	Середні
4	Складність реалізації	Низька	Висока	Висока

Значну роль у реалізації методів виявлення кіберзагроз відіграють системи виявлення та запобігання вторгненням (IDS/IPS). Вони аналізують мережевий трафік у режимі реального часу та дозволяють ідентифікувати підозрілу активність. Кращі результати досягаються при інтеграції таких систем із централізованими платформами моніторингу, такими як SIEM, що забезпечують кореляцію подій безпеки з різних джерел [4].

Сучасні підходи активно використовують технології машинного навчання, які дозволяють автоматизувати процес аналізу великих обсягів даних. Алгоритми класифікації, кластеризації та нейронні мережі забезпечують більш точне виявлення складних атак, зокрема багатовекторних і розподілених. Однак їх застосування пов’язане з високими вимогами до обчислювальних ресурсів та необхідністю якісних навчальних вибірок [5].

Таким чином, ефективність систем виявлення кіберзагроз визначається вибором відповідної архітектури, яка повинна враховувати баланс між точністю, швидкодією та складністю реалізації.

У роботі проаналізовано основні методи виявлення кіберзагроз у комп'ютерних мережах та визначено їх ключові характеристики. Встановлено, що сигнатурні методи є ефективними для відомих атак, тоді як аномалійні підходи дозволяють виявляти нові загрози. Найбільш перспективним є використання гібридних систем, які поєднують переваги обох підходів. Подальший розвиток цієї галузі пов'язаний із впровадженням технологій штучного інтелекту та вдосконаленням алгоритмів аналізу мережевих даних.

Список використаних джерел:

1. Stallings W. Network Security Essentials: Applications and Standards. 6th ed. Pearson, 2017. 816 p.
2. Scarfone K., Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf (дата звернення: 18.03.2026).
3. Behl A., Behl K. Cybersecurity and Cyberwar: What Everyone Needs to Know. Oxford University Press, 2017. 320 p.
4. Garfinkel S., Spafford G. Practical UNIX and Internet Security. 3rd ed. O'Reilly Media, 2003. 988 p.
5. Sommer R., Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. IEEE Symposium on Security and Privacy, 2010. URL: https://ieeexplore.ieee.org/document/5504793 (дата звернення: 18.03.2026).

УДК 004.738.5

ЕНЕРГОЗБЕРЕЖЕННЯ У ХМАРНИХ ТЕХНОЛОГІЯХ: ЕФЕКТИВНІСТЬ ТА ЕКОЛОГІЧНІСТЬ

Анголюк Ю. Д.
angoliukura@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Центри обробки даних забезпечують обробку, зберігання та передачу величезних обсягів даних. Однак їх діяльність має значний вплив ще й на навколишнє середовище, що викликає потребу в пошуку ефективних рішень для зниження цього впливу.

За даними Міжнародного агентства енергетики ще у 2020 році дата-центри споживали приблизно 1% від глобального споживання електроенергії. У своїх звітах вони запевняють, що цей відсоток буде зростати і може досягти позначки 8% до 2030 року. Це обумовлено низкою факторів, зокрема активним використанням онлайн-сервісів, збільшенням обсягу даних та поширенням хмарних обчислень.

Впровадження технології Інтернету речей в інфраструктуру центрів обробки даних (ЦОД) є надзвичайно актуальною темою в умовах зростання обсягів оброблюваної інформації, енергоспоживання та розвитку хмарних і AI-сервісів [3].

ЦОД витрачають значні обсяги електроенергії для живлення серверів, систем зберігання даних та охолодження обладнання [1].

Дата-центри споживають приблизно 1-2% світової електроенергії, і очікується, що ця цифра зростатиме у міру прискорення цифрової трансформації [4].

Одними з найбільш енерговитратних компонентів є системи охолодження та живлення. На них може припадати до 50% загального споживання енергії [1].

Однією з ключових сфер, на яку потрібно звернути увагу, є ефективність охолодження. Центри обробки даних виділяють значну кількість тепла завдяки

високому розміщенню обладнання та безперервній роботі. Завдяки оптимізації систем охолодження оператори центрів обробки даних можуть значно знизити споживання енергії.

Впровадження таких технологій, як утримання гарячого/холодного коридору, точне охолодження та рішення для керування повітряним потоком, може покращити ефективність охолодження, мінімізувати витрати енергії та підвищити загальну продуктивність [2].

Щоб зменшити цей вплив, дата-центри використовують більш ефективні технології охолодження замість повітряного. Наприклад, рідинне або випаровувальне [1].

Консолідація та віртуалізація є додатковими підходами до підвищення енергоефективності. Консолідуючи сервери, системи зберігання та мережеве обладнання, центри обробки даних можуть усунути недовикористані ресурси та зменшити енергоспоживання.

Технології віртуалізації дозволяють ефективно розподіляти ресурси, дозволяючи кільком віртуальним машинам працювати на одному фізичному сервері. Це призводить до вищого рівня використання сервера та значної економії енергії [2].

Віртуалізація серверів дозволяє запускати кілька віртуальних машин на одному фізичному сервері, що зменшує потребу використання додаткових серверів, необхідних для ефективної роботи. Власне, у такий спосіб знижується загальна потреба в електроенергії [1].

Інтеграція IoT дозволяє будувати багаторівневі архітектури, що включають розподілені сенсорні мережі, інтелектуальну аналітику та автоматизоване управління інфраструктурою в цілому.

У роботі проаналізовано типову архітектуру, роль ключових компонентів (сенсорів, актуаторів, шлюзів, edge-вузлів), а також узагальнено практичний досвід провідних компаній щодо оптимізації енергоспоживання та підвищення стійкості інфраструктури.

Показано, що використання IoT-компонентів сприяє досягненню значущих показників за ключовими метриками KPI: зниження середнього часу виявлення та усунення відхилень, покращення енергоефективності, підвищення точності виявлення аномалій [3].

Перехід на використання відновлювальних джерел енергії, таких як сонячні панелі та вітрові турбіни, може значно знизити вуглецевий слід. Вони генерують електроенергію з використанням безпечних для навколишнього середовища джерел: сонячного випромінювання та вітру, тому не супроводжуються викидами CO_2 в атмосферу.

Підвищуючи енергоефективність, дата-центри можуть зменшити свій вуглецевий слід і сприяти глобальним зусиллям у боротьбі зі зміною клімату. Ініціативи, такі як використання енергоефективних систем охолодження, оптимізація використання електроенергії та впровадження відновлюваних джерел енергії, є ключовими для мінімізації впливу на довкілля.

ЦОД також впроваджують інноваційні методи управління водними ресурсами, що дозволяє знижувати витрати води на 70% у порівнянні з традиційними методами. Наприклад, застосування закритих систем охолодження з можливістю повторного використання води.

Закриті системи використовують воду, яка циркулює в замкнутому контурі для охолодження обладнання. Це значно знижує споживання води, оскільки вона не випаровується і не скидається у стічні води [1].

Список використаних джерел:

1. ЦОД і екологія: сучасні технології, які впливають на наше майбутнє. *Gigacenter.ua*. URL: https://gigacenter.ua/ua/news/cod-ekolog-ya-suchasn-tehnolog-yak-vplivayut-na-nashe-maybutn (дата звернення: 31.03.2026).
2. Стратегії енергозбереження для покращення продуктивності центру обробки даних. *Opticomfiber.com*. URL: https://ua.opticomfiber.com/info/energy-saving-strategies-for-improved-data-cen-83151867.html (дата звернення: 31.03.2026).

3. Довженко Н., Іваніченко Є., Аушева Н., Шевчук Ю., Луковський Т. Дослідження архітектури дата-центрів з інтеграцією IoT-компонентів для забезпечення енергоефективності та кіберстійкості. *Електронне фахове наукове видання Кібербезпека: освіта, наука, техніка*. 2025. Т. 4, № 28. С. 547–564. URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/835 (дата звернення: 31.03.2026).
4. Цифровий апетит, який недооцінюють: правда про енергоспоживання ЦОД. *Denovo.ua*. URL: https://denovo.ua/blog/energy-dc-metrix-2025 (дата звернення: 31.03.2026).

УДК 004.738.5

БЕЗПЕКА ТА ЗАХИСТ БЕЗДРОТОВИХ МЕРЕЖ: ЗАГРОЗИ ТА МЕТОДИ ПРОТИДІЇ

Безчасний М. С.
hazardloka112@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Бездротові мережі сьогодні є невід’ємною частиною сучасної інформаційної інфраструктури, що активно використовується у навчальних закладах, організаціях та побуті. Основною перевагою таких мереж є мобільність і зручність доступу до ресурсів без використання кабелів. Водночас зростання їх популярності супроводжується підвищенням кількості загроз інформаційній безпеці, що обумовлює необхідність дослідження вразливостей і методів захисту бездротових мереж [1].

Бездротові мережі, зокрема технологія Wi-Fi, забезпечують швидкий обмін даними та підтримують підключення великої кількості пристроїв, однак передача інформації через радіоканал створює передумови для її перехоплення [4]. Це зумовлює появу різноманітних атак, спрямованих на порушення конфіденційності та цілісності даних.

Однією з основних загроз є несанкціонований доступ до мережі, що може виникати через слабкі паролі або відсутність шифрування. У таких випадках зловмисники можуть отримати доступ до конфіденційної інформації користувачів [2].

Серед найбільш поширених атак виділяють перехоплення трафіку, підбір паролів, атаки типу «людина посередині» (Man-in-the-Middle) та створення підроблених точок доступу [5]. Перехоплення трафіку дозволяє отримувати передані дані, тоді як атаки підбору паролів базуються на автоматизованому переборі комбінацій. Особливо небезпечними є атаки з використанням фальшивих точок доступу, що вводять користувачів в оману.

Для забезпечення безпеки бездротових мереж застосовуються сучасні методи захисту, зокрема використання протоколів шифрування WPA2 та WPA3, які забезпечують надійний рівень захисту переданих даних. Важливим є також правильне налаштування мережевого обладнання, зокрема зміна стандартних паролів, оновлення програмного забезпечення та обмеження доступу до мережі [5].

Додатковими заходами безпеки є використання фільтрації MAC-адрес та систем моніторингу мережі, які дозволяють виявляти підозрілу активність і своєчасно реагувати на загрози. Аналіз вразливостей мережі здійснюється за допомогою спеціалізованих інструментів, що дозволяють оцінити рівень захисту та визначити потенційні ризики.

Отже, бездротові мережі є важливою складовою сучасної інформаційної інфраструктури, однак вони залишаються вразливими до різноманітних кіберзагроз. Забезпечення їх безпеки потребує комплексного підходу, що включає використання сучасних методів шифрування, належне налаштування обладнання та постійний моніторинг мережі. Реалізація цих заходів дозволяє суттєво знизити ризики несанкціонованого доступу та забезпечити захист інформації користувачів [3].

Список використаних джерел:

1. Гізун А. І., Гнатюк С. О., Щербина В. П. Сучасні технології захисту інформації в бездротових мережах: навч. посіб. Київ: НАУ, 2021. 240 с.
2. Бурячок В. Л., Корченко О. Г., Ткач Ю. М. Інформаційна та кібербезпека: соціотехнічний аспект: підручник. Київ: ДУТ, 2022. 352 с.
3. OpenAI. GPT-4 Technical Report. 2023. URL: https://openai.com/research (дата звернення: 19.03.2026).
4. Комп'ютерні мережі: підручник / А. Г. Микитишин та ін. Тернопіль: ТНТУ ім. І. Пулюя, 2020. 256 с.
5. Жебка В. В., Сєрих С. О. Аналіз методів автентифікації та шифрування у бездротових мережах стандарту IEEE 802.11. Кібербезпека: освіта, наука, техніка. 2023. № 2(14). С. 45–58.

УДК 004.8:81'243

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У ВИВЧЕННІ АНГЛІЙСЬКОЇ МОВИ

Прозоровська І.М.
zorovarina08@gmail.com
Черкаський державний фаховий бізнес-коледж
м. Черкаси, Україна

Поява інтернету декілька десятиліть тому змінила життя людей у багатьох сферах. Згодом, завдяки розвитку «міжнародної мережі», був створений штучний інтелект (ШІ), який набув своєї максимальної актуальності за останні декілька років. У вивченні іноземних мов, штучний інтелект на сьогодні стає необхідним потужним помічником, як для студентів, так і для викладачів.

Серед найважливіших складових у вивченні мови, можна виділити такі, як зворотній зв'язок (feedback), поступове навантаження у навчання (scaffolding), практика відтворення (retrieval practice) й інтервальне повторення (spaced practice).

Зворотній зв'язок – це інформація, яку отримують учні щодо їх актуального засвоєння матеріалу, викриваючи пробіли у засвоєному матеріалі з огляду на навчальну мету. Ключова характеристика фідбеку, яка відрізняє його від стандартного оцінювання, – наявність опису однакової кількості гарних і негативних рис [1]. Для зворотнього зв’язку можна використовувати наступні інструменти ШІ - Grammarly, ProWritingAid, ChatGPT/Gemini, Socratic(Google).

Поступове навантаження у навчанні (scaffolding). Стратегія скаффолдингу є однією з ефективних стратегій взаємодії в ході самостійної роботи студентів. Підтримка навчання при стратегії скаффолдингу полягає у наданні ресурсів, завдань відповідного рівня, прикладів, зразків, інструкцій та віддалене керівництво самостійною роботою студентів. Матеріали для навчання надаються послідовно протягом процесу, який супроводжується контролем з боку викладача [2]. Єфективні ШІ засоби для скаффолдінгу – ChatGPT/Gemini (структуровані промти), MagicSchool.ai, Read&Write (TextHelp) (для читання), Canva Magic Design (для візуальних скаффолдів), Text Rewriting Tools (QuilBot /WriteSonar) (для рівневих текстів).

Практика відтворення (retrieval practice) передбачає відтворення у пам’яті вивченого матеріалу без опори на джерела. Це стратегія перенесення інформації у свідомість, яка сприяє покращенню та прискоренню навчання [3].

Для цієї практики стануть у пригоді такі ШІ інструменти, як Quizlet (для створення flashcards), Kahoot! (генератор запитань), QuestionWell (генератор тестів), Duolingo Max (створення завдань на рольові ігри).

Практика інтервального повторення (spaced practice) – це повторення вивченного матеріалу через певні проміжки часу. Ця практика дає нам змогу переривати процес забування у учнів й підтримувати збереження знань на довготривалий період [4]. Допоможуть у практиці цього методу наступні ШІ засоби: Anki, Memrise, Quizlet Long-Term Learning, EdApp.

Завдяки появі штучного інтелекту процес викладання та вивчення англійської мови став набагато продуктивним та цікавим. Оскільки технології штучного інтелекту постійно розвиваються, все більше нових продуктів

створюється. Задача як викладачів, так і студентів опановувати ці продукти і навчатися промптингу для ефективнішого їх використання.

Список використаних джерел:

1. Сучасне викладання іноземних мов. Що таке feedback та як його давати. URL: https://nus.org.ua/2018/10/18/suchasne-vykladannya-inozemnyh-mov-shho-take-feedback-ta-yak-jogo-davaty/ (Дата звернення: 08.04.2026)
2. Скаффолдинг як комунікативна стратегія взаємодії в ході самостійної роботи студентів у процесі вивчення англійської мови. URL: http://dspace-s.msu.edu.ua:8080/bitstream/123456789/8164/1/Scaffolding_%20as_%20a_%20communicative%20_strategy_%20of_%20interaction_%20in_%20the_%20course_%20of%20_independent%20_work.pdf (Дата звернення: 08.04.2026)
3. What is retrieval practice? URL: https://www.retrievalpractice.org/why-it-works (Дата звернення: 08.04.2026)
4. An introduction to spaced practice URL: https://my.chartered.college/early-career-hub/an-introduction-to-spaced-practice/ (Дата звернення: 08.04.2026)

УДК 004.032.26

МУЗИКА, СТВОРЕНА ШТУЧНИМ ІНТЕЛЕКТОМ: ЧИ МОЖЕ ШІ ЗАМІНИТИ КОМПОЗИТОРІВ?

Затяміна М. І.
maya.zatyamina@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Музика, створена за допомогою технологій штучного інтелекту, стрімко переходить із експериментальної сфери у практичне застосування. Сучасні алгоритми здатні аналізувати значні обсяги музичних даних, виявляти закономірності гармонії, ритму та структури й на цій основі генерувати нові композиції [1].

Застосування методів машинного навчання, зокрема нейронних мереж, дозволяє імітувати стиль окремих композиторів або створювати оригінальні

музичні твори, що зумовлює необхідність наукового осмислення ролі штучного інтелекту у творчих процесах.

Однією з ключових переваг алгоритмів генерації музики є їх висока швидкість та масштабованість. Штучний інтелект здатний створювати значну кількість варіацій музичних композицій за короткий проміжок часу, а також адаптувати їх відповідно до конкретних потреб, зокрема у сфері відеоігор, кінематографу та рекламної індустрії [2].

Крім того, алгоритмічні системи не піддаються фізичній втомі та не мають творчих криз, що дозволяє їм ефективно комбінувати різні музичні стилі та створювати нові художні рішення. Це сприяє розширенню можливостей музичного експерименту та демократизації процесу створення музики [3].

Водночас людська творчість характеризується унікальними властивостями, які складно повністю відтворити засобами штучного інтелекту. Композитори інтегрують у музичні твори власний досвід, емоції, культурні особливості та інтуїтивні рішення. Натомість штучний інтелект функціонує на основі вже існуючих даних, що обмежує його здатність до створення принципово нових ідей [1].

Важливим аспектом є також питання авторства та етичних норм. У випадку створення музики за допомогою алгоритмів виникає проблема визначення суб’єкта авторського права: розробника програмного забезпечення, користувача чи системи штучного інтелекту. Додатково актуалізується питання використання навчальних даних, оскільки моделі часто тренуються на існуючих музичних творах без прямого дозволу їх авторів, що викликає дискусії щодо дотримання прав інтелектуальної власності [3].

Попри значні досягнення у сфері генерації музики, доцільно розглядати штучний інтелект не як заміну композитора, а як інструмент підтримки творчого процесу. Ефективною є модель взаємодії, у якій штучний інтелект генерує ідеї, а людина здійснює їх відбір, інтерпретацію та наповнення змістом [2].

Отже, штучний інтелект вже здатний створювати музичні композиції високого рівня, однак його роль залишається переважно допоміжною. Людська

творчість зберігає ключове значення як джерело емоційної насиченості, смислової глибини та культурного контексту, тоді як штучний інтелект виступає потужним інструментом, що трансформує підходи до створення та сприйняття музики.

Список використаних джерел:

1. Музика і штучний інтелект. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Музика_і_штучний_інтелект (дата звернення: 23.03.2026).
2. ШІ для створення музики: нові можливості композиторів і артистів. URL: https://88000.com.ua/shi-dlya-stvorennya-muzyky-novi-mozhlyvosti-kompozytoriv-i-artystiv/ (дата звернення: 23.03.2026).
3. ШІ та музика: хто насправді виграє? URL: https://expert.com.ua/175929-shi-ta-muzyka-hto-naspravdi-vygraye.html (дата звернення: 23.03.2026).

УДК 004.8:159.9.072

АВТОМАТИЗАЦІЯ ТА ПРИХОВАНЕ НАВАНТАЖЕННЯ ЛЮДИНИ В ЕПОХУ ШТУЧНОГО ІНТЕЛЕКТУ

Дрожаний Є.О.
veyryff4@gmail.com
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м. Черкаси, Україна

Сьогодні ставлення до штучного інтелекту зазвичай коливається між двома крайнощами: страхом втратити роботу та надією на те, що алгоритми вирішать усі проблеми з продуктивністю. Штучний інтелект – це не просто надзвичайно потужний інструмент. Справжня складність полягає не стільки в самій автоматизації, скільки в тому, що штучний інтелект непомітно змінює рівень навантаження людини. Надмірне й некритичне впровадження автоматизованих рішень впливає на психіку саме там, де це найменше очікується.

Довгий час вважалося, що якщо автоматизувати рутину, у людей звільниться час для творчості. Але виявилося, що рутина була необхідною. Механічні задачі, заповнення таблиць чи сортування листів, працювали як когнітивний перепочинок. Поки людина автоматично робила звичну роботу, мозок «перезавантажувався». Зараз же, коли штучний інтелект допомагає вирішувати прості задачі, кожна хвилина роботи вимагає максимальної концентрації та прийняття рішень. Людина не здатна бути креативною 8 годин поспіль [1]. Результатом стає «AI brain fry» – стан глибокої ментальної втоми, туману в голові та втрати здатності до концентрації, спричинений надмірним розумовим навантаженням без природних пауз [2]. Останні дослідження показують парадоксальну річ: коли ШІ повністю забирає так звану «рутину», рівень вигорання серед працівників не падає, а навпаки стрімко зростає [3].

Ситуацію погіршує те, що генеративні моделі схильні до «галюцинацій». Вони видають помилки з дуже впевненим виглядом. Через це фахівці перетворилися на наглядачів: замість того, щоб створювати щось своє, вони годинами вичитують згенерований текст чи код, шукаючи приховані неточності [4]. Психологи називають цей стан «податком на пильність» (vigilance tax). Парадокс полягає в тому, що моніторинг розумної, але схильної до помилок автоматизованої системи часто вимагає значно більше ментальної енергії, ніж самостійне виконання тієї ж роботи. Знайти витончену логічну хибу в чужому алгоритмі об'єктивно важче і нудніше, ніж написати свій з нуля. У результаті люди починають відчувати себе просто «прибиральниками за штучним інтелектом», що повністю вбиває будь-яке задоволення від роботи і відчуття власної професійної значущості [5].

Додається ще й психологічний тиск. Людина підсвідомо порівнює себе не з іншою людиною, а з машиною, яка не втомлюється і видає результати миттєво. Працівники починають вимагати від себе такої ж безперебійності, що призводить до синдрому самозванця, забуваючи, що цінність людини полягає не у швидкості генерації символів [6]. Проте корпоративна гонка за надмірною ефективністю змушує людей сприймати себе як «повільні й дефектні

алгоритми», що стає серйозним викликом для ментального здоров'я цілого покоління спеціалістів.

Таким чином, штучний інтелект слід розглядати як багаторівневу технологію, яка вимагає чітких регламентів використання та інтеграції у робочі процеси з урахуванням біологічних і психологічних особливостей людини. Необхідно зберігати частину рутинних завдань людини як елемент когнітивної гігієни, а також розробляти протоколи для безпечної роботи з генеративними моделями. Лише за умови балансу між технологічними можливостями та людськими ресурсами можна забезпечити стійку продуктивність і зберегти психічне здоров'я працівників у новій цифровій реальності.

Список використаних джерел:

1. Bedard J., Kropp B. The AI Productivity Paradox: Cognitive Overload and «Brain Fry» in the Workplace. *Harvard Business Review*. 2026. URL: https://hbr.org/ (дата звернення: 02.03.2026).
2. Montes R., Khojah M. Cognitive Offloading and the Impact of LLMs Usage on IT Professionals. *Journal of Software Engineering for Robotics*. 2025. Vol. 15, No. 2. P. 45–62.
3. Інтенсивне використання ШІ викликає когнітивну втому у працівників – дослідження. Vector. URL: https://vctr.media/ua/intensyvne-vykorystannya-shi-vyklykaye-kognityvnu-vtomu-u-praczivnykiv-doslidzhennya-320842 (дата звернення: 03.03.2026).
4. Henry S. The Hidden Emotional Cost of Artificial Intelligence and Human-Machine Comparison. *Frontiers in Psychology*. 2025. Vol. 16. P. 112–128.
5. Bankins S., Formosa P. Making Artificial Intelligence Work at Work: The Role of Human Resource Practices. Journal of Business Ethics. 2026. Vol. 182. P. 15–32.
6. Yegge S. Artificial intelligence and employee well-being: the crucial role of self-efficacy in avoiding burnout. Psychology Research and Behavior Management. 2024. Vol. 17. P. 230–245.

УДК 004.8

ШТУЧНИЙ ІНТЕЛЕКТ І МОРАЛЬ: ЧИ МОЖЛИВО ВИХОВАТИ «ЕТИЧНИЙ» AI?

Карабань Р. Є.
karabas736901.11@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Технології швидко розвиваються, і штучний інтелект (AI) став важливою частиною життя, знаходячи застосування у медицині, транспорті, фінансах і повсякденних цифрових сервісах [1]. Однак разом із поширенням AI виникає питання його етичності, оскільки такі системи дедалі частіше приймають рішення, що впливають на життя людей. У зв'язку з цим проблема відповідальності, справедливості та безпечного використання штучного інтелекту набуває особливої актуальності.

У сучасному світі сформувалася окрема галузь досліджень – етика штучного інтелекту (AI Ethics), яка визначає принципи безпечного, прозорого та відповідального використання технологій [2]. Над її розвитком працюють фахівці різних галузей, що підкреслює міждисциплінарний характер проблеми.

Однією з ключових проблем є упередженість алгоритмів, що виникає через використання даних, які можуть містити помилки або соціальні стереотипи. Це може призводити до дискримінаційних рішень, зокрема у сфері працевлаштування або аналізу соціальних даних [3]. Подібні ризики спостерігаються і у фінансовій сфері, де системи кредитного скорингу можуть створювати нерівні умови доступу до ресурсів.

Важливим напрямом розвитку є використання explainable AI, що дозволяє пояснювати рішення систем і підвищує рівень довіри до них [5]. У критичних сферах, таких як транспорт і медицина, застосовується принцип контролю людини та мінімізації потенційної шкоди [4].

У фінансовій сфері AI використовується для кредитного скорингу. Проблема полягає в тому, що модель може враховувати неочевидні фактори,

наприклад район проживання або тип витрат, що призводить до нерівного доступу до кредитів. Тому впроваджуються explainable AI-системи, які пояснюють причини рішень (рівень доходу, кредитна історія, боргове навантаження).

Ще один важливий аспект – безпека використання. У безпілотних автомобілях системи постійно тестуються в різних умовах: дощ, туман, нічний час. У критичних ситуаціях алгоритми діють за заздалегідь визначеними принципами – мінімізація шкоди, дотримання правил і захист життя.

Складною проблемою є також питання відповідальності. Якщо система штучного інтелекту допускає помилку, складно визначити, хто саме винен: розробник, компанія чи користувач. Це особливо актуально у випадках медичних систем.

У медицині важливу роль відіграє якість даних. Якщо модель навчена переважно на даних однієї групи пацієнтів, вона може гірше працювати для інших. Тому системи проходять клінічну перевірку на різних вибірках. При цьому остаточне рішення завжди залишається за лікарем: AI лише допомагає, а не замінює спеціаліста.

У соціальних мережах рекомендаційні алгоритми можуть підсилювати дезінформацію або радикальний контент, оскільки орієнтуються на залученість користувачів. Для вирішення цього застосовують обмеження: зниження видимості недостовірного контенту, його маркування та модерацію.

У сфері безпеки системи розпізнавання облич можуть мати різну точність для різних груп людей, що призводить до помилкових ідентифікацій. Тому такі технології додатково тестуються, обмежуються або регулюються.

В освіті системи автоматичного оцінювання можуть бути упередженими до стилю відповідей або мови. Тому їх використовують лише як допоміжний інструмент, а не як остаточне джерело оцінки.

Для створення етичного штучного інтелекту застосовують кілька основних підходів:

- Програмування етичних правил – у системи закладають обмеження та правила поведінки. Наприклад, чат-боти не відповідають на небезпечні або шкідливі запити та уникають мови ненависті.
- Використання якісних і перевірених даних – моделі навчаються на достовірній і різноманітній інформації, щоб зменшити упередженість.
- Контроль і перевірка людиною – системи тестуються та коригуються перед впровадженням.

Проблема відповідальності за рішення AI залишається відкритою, оскільки складно визначити, хто несе відповідальність за помилки системи – розробник, організація чи користувач [2]. У соціальних мережах алгоритми можуть сприяти поширенню дезінформації, що потребує впровадження механізмів регулювання та контролю [3].

Для забезпечення етичності штучного інтелекту застосовуються такі підходи: використання якісних даних, програмування етичних обмежень, аудит алгоритмів, забезпечення прозорості рішень та участь людини у процесі прийняття рішень (human-in-the-loop) [4].

Штучний інтелект є потужним інструментом розвитку суспільства, однак його використання супроводжується низкою етичних викликів. Забезпечення справедливості, прозорості та безпеки AI-систем можливе лише за умови комплексного підходу, що поєднує технічні рішення та людський контроль. Водночас штучний інтелект не має власної моралі, тому відповідальність за його застосування повністю покладається на людину [1].

Список використаних джерел:

1. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 4th ed. Pearson, 2021. 1136 p.
2. Floridi L. Ethics of Artificial Intelligence. Oxford: Oxford University Press, 2023. 320 p.
3. Analytics Steps. AI Ethics: Addressing Bias and Fairness in Machine Learning Models. URL: https://www.analyticssteps.com (дата звернення: 18.03.2026).

4. European Commission. Ethics Guidelines for Trustworthy AI. URL: https://digital-strategy.ec.europa.eu (дата звернення: 18.03.2026).
5. IBM. AI Ethics. URL: https://www.ibm.com/artificial-intelligence/ethics (дата звернення: 18.03.2026).

УДК 004.8:004.932

РОЗРОБКА ФОТОРЕАЛІСТИЧНОГО ЦИФРОВОГО АВАТАРА З ВИКОРИСТАННЯМ ГЕНЕРАТИВНИХ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ

Компанієць Ю. М.
yuliannakompaniets@gmail.com
Черкаський державний фаховий бізнес–коледж
Ночевнов Д. П.
м. Черкаси, Україна

Цифрові аватари сьогодні активно використовуються у відеоконференціях, онлайн-освіті, іграх, медичних тренажерах і середовищах розширеної реальності, а потреба у якісних і реалістичних аватарах постійно зростає. Якщо раніше їх створення ґрунтувалося переважно на трудомісткому ручному 3D-моделюванні, то поява генеративних моделей штучного інтелекту дала змогу автоматично синтезувати фотореалістичні обличчя та змінювати їх параметри на основі лише кількох вхідних фотографій.

GAN стали першою широко застосованою архітектурою для генерації реалістичних зображень, а StyleGAN – одним із найвідоміших рішень для синтезу облич із високою якістю та керованістю стилем [1]. Variational Autoencoder (VAE) реалізують інший підхід через латентний простір, поступаючись GAN у деталізації, але забезпечуючи стабільніше навчання та зручну інтерполяцію. Отже, GAN і VAE можна розглядати як недифузійні генеративні підходи, що становлять теоретичну основу для порівняння із сучасними дифузійними моделями. Дифузійні моделі виконують генерацію шляхом поетапного видалення шуму, а Latent Diffusion Model, на якій побудований Stable Diffusion, переносить цей процес у стиснутий латентний простір, що пришвидшує обчислення [2]. Додатково ControlNet надає таким

моделям можливість точно керувати просторовою структурою зображення, зокрема позою, контурами обличчя та напрямком погляду, що робить їх ефективним інструментом для синтезу фотореалістичних аватарів [3].

Методи повноцінної 3D-реконструкції, зокрема підходи на основі параметричних 3D-моделей обличчя, таких як FLAME, і сучасні методи рендерингу, зокрема 3D Gaussian Splatting, потребують значних обчислювальних ресурсів і великого обсягу вхідних даних. Внаслідок цього їх використання в практичних сценаріях є ускладненим. З огляду на це в роботі обрано підхід на основі 2D-генеративних моделей, який забезпечує отримання фотореалістичних результатів навіть за обмеженої кількості вхідних зображень.

У практичній частині роботи обробку вхідних фотографій реалізовано із використанням програми в середовищі Google Colaboratory, тоді як генерацію фотореалістичного аватара та отримання фінального результату виконано за допомогою сервісу Ip-Adapter-FaceID, доступного як Hugging Face Space [6]. Процес обробки можна умовно поділити на чотири основні етапи.

На першому етапі виконується детекція обличчя та його вирівнювання. Бібліотека OpenCV у поєднанні з детектором на основі MTCNN або RetinaFace виявляє обличчя на вхідному зображенні, визначає ключові точки (очі, ніс, кути рота) та виконує афінне перетворення, приводячи обличчя до стандартного положення, як на рисунку 1.

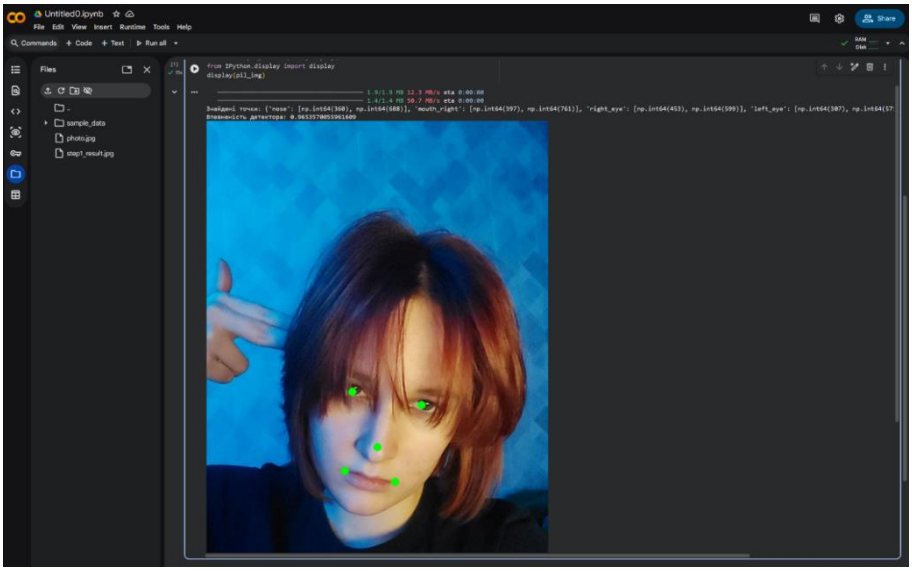


Рисунок 1 – Детекція обличчя

На другому етапі з підготовленого зображення витягується вектор ідентичності за допомогою попередньо навченої моделі розпізнавання облич – зокрема, ArcFace. Цей вектор містить компактне числове представлення унікальних рис обличчя, яке надалі використовується як умова при генерації, щоб забезпечити схожість аватара з реальною особою (див. рисунок 2).

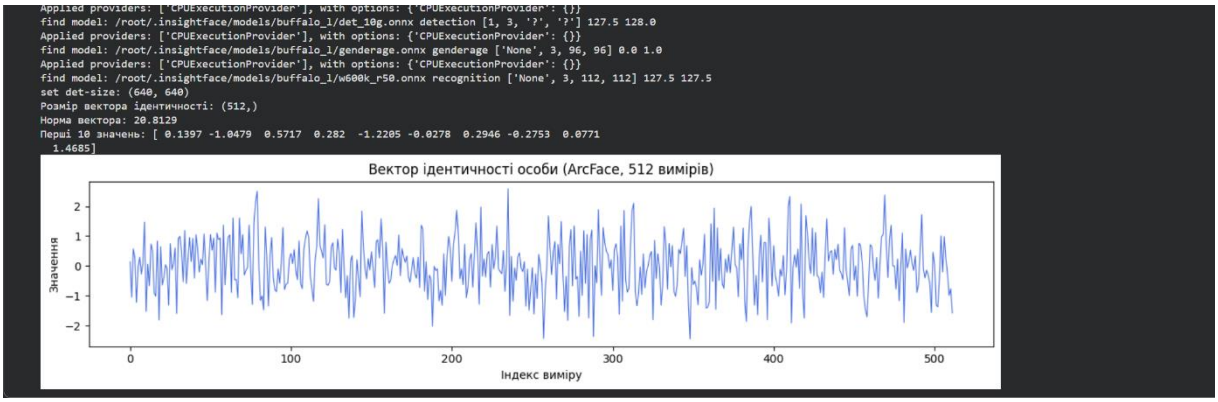


Рисунок 2 – Вектор ідентичності

На третьому етапі відбувається безпосередня генерація аватара за допомогою Stable Diffusion із підключеним ControlNet, який отримує умовне зображення у вигляді карти глибини або скелету обличчя, тоді як текстовий промпт задає бажаний стиль, освітлення та ракурс (рисунки 3, 4). Вектор ідентичності інтегрується через IP-Adapter, запропонований в [5], що дає змогу передавати риси обличчя без донавчання моделі для кожної нової особи.

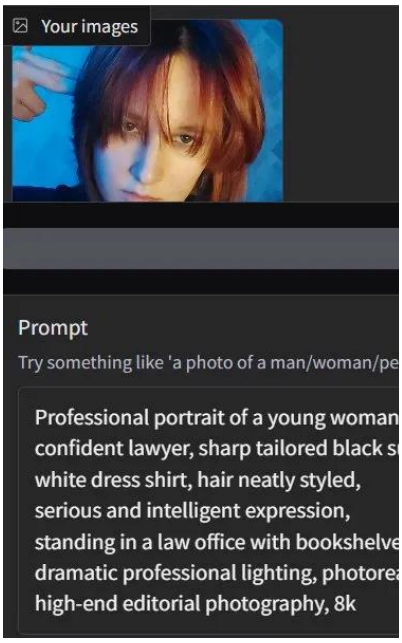


Рисунок 3 – Вхідні фото та промт

Рисунок 4 – Згенерований результат

На четвертому етапі виконується постобробка: підвищення роздільності за допомогою ESRGAN, корекція кольорів, а також опціональне зменшення артефактів за допомогою GFPGAN.

Використаний підхід демонструє практично прийнятну якість за метриками FID, LPIPS та ID Similarity, який отримав позитивну суб'єктивну оцінку від учасників тестування. Технологічний стек реалізації включає Python 3.11, PyTorch 2.x, Hugging Face Diffusers, OpenCV і CUDA, а всі компоненти упаковано у Docker-образ для спрощення розгортання.

Серед ключових напрямків подальшого розвитку – адаптація системи до генерації у реальному часі, розширення функціональності для підтримки анімації аватара на основі мімічного відеосигналу, а також дослідження мультимодальних підходів, що поєднують зображення, звук і текст для ще більш природної взаємодії.

Список використаних джерел

1. Karras T., Laine S., Aila T. A Style–Based Generator Architecture for Generative Adversarial Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021. Vol. 43, No. 12. P. 4217–4228.

2. Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B. High–Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF CVPR. 2022. P. 10684–10695.
3. Zhang L., Rao A., Agrawala M. Adding Conditional Control to Text–to–Image Diffusion Models. Proceedings of the IEEE/CVF ICCV. 2023. P. 3836–3847.
4. Kerbl B., Kopanas G., Leimkühler T., Drettakis G. 3D Gaussian Splatting for Real–Time Radiance Field Rendering. ACM Transactions on Graphics. 2023. Vol. 42, No. 4. Article 139.
5. Ye H., Zhang J., Liu S., Han X., Yang W. IP–Adapter: Text Compatible Image Prompt Adapter for Text–to–Image Diffusion Models. arXiv preprint. 2023. arXiv:2308.06721. URL: https://doi.org/10.48550/arXiv.2308.06721, (дата звернення: 20.03.2026).
6. IP-Adapter-FaceID Demo. URL: https://multimodalart-ip-adapter-faceid.hf.space (дата звернення: 20.03.2026).

УДК 004.891

ШТУЧНИЙ ІНТЕЛЕКТ ЯК ІНСТРУМЕНТ ОПТИМІЗАЦІЇ АЛГОРИТМІВ ОБРОБКИ ДАНИХ

Кононенко С.А.
vinsofia43@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В.Ю.
м. Черкаси, Україна

Сучасний етап розвитку глобальної інформаційної інфраструктури характеризується стрімким переходом від кількісного накопичення даних до необхідності їх якісної та надшвидкої обробки. Традиційні детерміновані алгоритми, що десятиліттями складали фундамент комп'ютерних наук, у сучасних умовах Big Data починають демонструвати обмеженість, зумовлену статичністю їхньої внутрішньої логіки. Проблема полягає в тому, що класичний алгоритм не враховує специфіку та внутрішню структуру конкретного набору даних, орієнтуючись лише на найгірший або середній теоретичний випадок

часової складності. Впровадження технологій штучного інтелекту, зокрема методів глибокого машинного навчання, дозволяє подолати цей бар'єр, перетворюючи жорстко задані інструкції на адаптивні механізми, здатні до динамічного підлаштування під контекст вхідної інформації.

Концептуальний підхід до інтелектуалізації обробки даних базується на створенні гібридних архітектур, де штучний інтелект виступає не заміною базового обчислювального процесу, а високоефективним диспетчером. Використання предиктивної аналітики дозволяє системі заздалегідь оцінювати статистичні характеристики інформаційного потоку. Наприклад, аналізуючи попередні запити та структуру вхідних пакетів, інтелектуальна надбудова може динамічно перерозподіляти навантаження між вузлами обробки, випереджаючи виникнення критичних затримок. Це забезпечує перехід від реактивного управління ресурсами до проактивного, що є критично важливим для систем реального часу та об’єктів критичної інфраструктури.

Особливого значення набуває ревізія фундаментальних алгоритмів сортування та індексації через призму навчених моделей. Традиційні структури, такі як B-дерева або хеш-таблиці, ігнорують розподіл ключів у пам'яті. Натомість інтелектуальні індексні структури (Learned Index Structures) використовують регресійні моделі для прогнозування точного розташування даних. Замість послідовного перебору рівнів ієрархічного дерева, система фактично «обчислює» адресу потрібного елемента. Такий підхід дозволяє суттєво зменшити кількість звернень до кеш-пам'яті та центрального процесора, що в масштабах центрів обробки даних трансформується у значну економію електроенергії та часу. Більше того, адаптивне сортування на базі нейронних мереж здатне самостійно обирати найбільш раціональний метод впорядкування (наприклад, QuickSort або MergeSort) залежно від ступеня попередньої деградації впорядкованості масиву.

Оптимізація процесів підготовки даних (ETL-процесів) за допомогою штучного інтелекту дозволяє розв'язати проблему семантичного розриву. У гетерогенних інформаційних системах дані часто мають різний формат та опис,

що раніше вимагало значних зусиль програмістів для написання скриптів трансформації. Сучасні інтелектуальні інструменти забезпечують автоматичне мапування полів на основі аналізу контексту та значень, що зберігаються в базах. Це доповнюється механізмами інтелектуальної детекції аномалій, які в реальному часі відсікають інформаційний «шум» та технічні похибки сенсорів, забезпечуючи високу якість даних на виході. Застосування автоенкодерів для стиснення інформації дозволяє передавати максимально корисний обсяг даних через обмежені канали зв'язку, що є особливо актуальним для мобільних систем та технологій індустріального інтернету речей.

Проте, розширення використання штучного інтелекту як інструменту оптимізації супроводжується низкою викликів, які потребують наукового осмислення. Передусім, це стосується проблеми інтерпретованості результатів («black box problem»). Коли алгоритм оптимізації приймає рішення на основі ваг нейронної мережі, стає важко верифікувати безпеку та стабільність такого рішення в критичних умовах. Також слід враховувати енергетичні витрати на навчання самих моделей. Якщо етап навчання потребує більше ресурсів, ніж економія, отримана від подальшої оптимізації, використання такого підходу є економічно недоцільним. Тому актуальним напрямом досліджень залишається пошук балансу між складністю інтелектуальної моделі та реальною вигодою від її впровадження в обчислювальний процес.

Підсумовуючи, можна стверджувати, що трансформація алгоритмів обробки даних у сторону їх інтелектуалізації є безальтернативним вектором розвитку IT-галузі. Це забезпечує не лише приріст продуктивності на 20–40%, а й створює фундамент для побудови когнітивних систем, здатних оперувати неструктурованою інформацією великих обсягів з ефективністю, що раніше була доступна лише вузькоспеціалізованим детермінованим системам. Майбутнє галузі лежить у площині створення легковагових нейромережевих моделей, які зможуть інтегруватися безпосередньо у логіку мікропроцесорів для апаратної підтримки інтелектуальної обробки даних.

Список використаної літератури:

1. Розпорядження Кабінету Міністрів України від 02 груд. 2020 р. № 1556-р про схвалення Концепції розвитку штучного інтелекту в Україні.
2. Глибоке навчання та його застосування в задачах обробки великих даних: монографія / О. В. Сирота та ін. Київ : Наукова думка, 2023. 245 с.
3. Рассел С., Норвіг П. Штучний інтелект: сучасний підхід. 4-те вид. / пер. з англ. Київ : Діяльність, 2021. 1152 с.
4. Kraska T., Alizadeh M., Beutel A. The Case for Learned Index Structures. Proceedings of the 2018 International Conference on Management of Data. 2018. P. 489–504.DOI:(https://doi.org/10.1145/3183713.3196909)
5. Оптимізація алгоритмів обробки даних засобами машинного навчання / І. П. Коваленко та ін. Системи обробки інформації. 2022. Вип. 2 (169). С. 45–52.

УДК 004.8

АВТОМАТИЧНА КЛАСИФІКАЦІЯ ФЕЙКОВИХ НОВИН ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ

Кріль О. М.
myrkotkot7@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

У сучасну епоху цифрової трансформації проблема поширення дезінформації та фейкових новин набула глобальних масштабів [2]. Величезний обсяг контенту, що генерується в соціальних мережах та онлайн-медіа, робить ручну перевірку фактів (fact-checking) малоефективною. Автоматична класифікація новин за допомогою штучного інтелекту (ШІ) стає критично важливим інструментом для підтримки інформаційної безпеки та медіагігієни.

Технологія автоматичної класифікації базується на методах обробки природної мови (NLP) та машинного навчання (ML). Процес аналізу зазвичай включає три основні етапи: попередню обробку тексту, вилучення ознак та

класифікацію. Попередня обробка передбачає токенізацію, видалення стоп-слів та лематизацію, що дозволяє зменшити інформаційний шум [1].

Для виявлення фейкового контенту алгоритми ШІ орієнтуються на кілька типів ознак:

- лінгвістичні (стилістика, емоційність, маніпулятивність);
- контекстуальні (джерело, автор, метадані);
- структурні (аналіз поширення інформації в мережі).

Для класифікації застосовуються як класичні методи (наївний байєсівський класифікатор, SVM), так і сучасні нейромережеві підходи. Найбільш ефективними є моделі-трансформери, зокрема BERT і RoBERTa, які дозволяють враховувати контекст і семантичні зв'язки між словами [4].

Однією з ключових переваг ШІ є можливість аналізу в режимі реального часу, що дозволяє виявляти фейковий контент до його масового поширення [3]. Крім того, використання Explainable AI (XAI) дає змогу пояснювати результати класифікації, підвищуючи довіру до систем [1].

Водночас автоматична класифікація стикається з низкою проблем. Зокрема, спостерігається постійне вдосконалення методів створення дезінформації, що формує ефект «гонки озброєнь» між системами захисту та зловмисниками [2]. Також складним залишається розмежування сатири та навмисної дезінформації.

Отже, автоматизація виявлення фейкових новин є необхідною складовою сучасної кібербезпеки. Поєднання технологій штучного інтелекту з експертною оцінкою людини забезпечує ефективний підхід до протидії дезінформації.

Список використаних джерел:

1. Гриценко О. М. Штучний інтелект в медіапросторі: виклики та можливості. Київ: Академія, 2024. 142 с.
2. Allcott H., Gentzkow M. Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*. 2017. Vol. 31. No. 2. P. 211–236.

3. Штучний інтелект як інструмент боротьби з дезінформацією. *Детектор медіа*. URL: https://detector.media (дата звернення: 18.03.2026).
4. Lazer D. J. et al. The science of fake news. *Science*. 2018. Vol. 359. P. 1094–1096.

UDC 004.8:004.932

DETECTION OF BEHAVIORAL STATES BASED ON SENSOR DATA USING MACHINE LEARNING

Levchenko. S.S.
lev4enkostanislav@gmail.com
Cherkasy State Business College
Ivanova. I.V.
Cherkasy, Ukraine

The presented work is dedicated to question classifications behavioral states participants based on analysis multimodal data from sensors. This study was implemented within the framework of a specialized English-language Python programming course [1]. A feature teaching was full integration technical English languages in education process: from study documentation machine learning libraries for communication with teachers and colleagues exclusively in English in the language for discussion architecture models and results experiments [2].

Development technologies Internet of Things and mobile devices allows collect big volumes data that much increases efficiency monitoring human activities in medicine, sports and security systems. At the same time complexity structures sensory data creates new challenges in the field processing information and recognition patterns.

In the conditions work on the project «CMI – Kaggle Competition» an analysis was conducted sequences data obtained from accelerometers, sensors rotation and time- of-flight (TOF) sensors. Main task was to predict four types behavior: execution gestures, movement towards the target, holding the hand near goals and state of rest [3].

Among main factors that provided high precision developed systems, you can to highlight the following:

- Using the Random algorithm Forest Classifier, which demonstrated noise immunity in indicators sensors.
- Thorough previous processing data, which included deletion irrelevant or empty values and balancing samples.
- Detection most significant signs, such as indicators orientations and temporary counters sequences.
- Application of multiclass metrics scores (Accuracy, F1-score) for analysis qualities models [4].

An important role in the implementation of the project was played by possession technical in English language, which allowed for effective use global resources, such as the Kaggle platform, and integrate modern deep learning techniques analysis data in your own development. According to the results of testing was achieved high equal accuracy classification (0.92), which confirms efficiency chosen one approach.

In this way you can make conclusion that combination skills Python development from free possession in English language in professional environment is a key factor in successful implementation complex IT projects. Effective recognition behavior requires a comprehensive approach that includes using modern machine learning technologies and constant improve professional competencies specialist.

References:

1. Child Mind Institute. Detect Behavior with Sensor Data. Access mode: https://www.kaggle.com/competitions/cmi-detect-behavior-with-sensor-data
2. Python Software Foundation. Random Forest Classification Documentation. Access mode: https://scikit-learn.org
3. Breiman L. Random Forests. – Machine Learning, 2001. Vol. 45. P. 5–32.
4. Standard Practice for Signal Processing in Sensor Systems. Gaithersburg: NIST, 2022.

УДК 004.8, 004.92

АНАЛІЗ АРХІТЕКТУР ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ 3D-ВІЗУАЛІЗАЦІЇ

Куций В. В.
vladkytsiy@gmail.com
Черкаський державний фаховий бізнес-коледж
Марченко С. В.
м.Черкаси, Україна

Сучасна індустрія комп’ютерної графіки та створення цифрового контенту (Digital Content Creation) перебуває на етапі фундаментальної технологічної трансформації [1; 2]. Ця зміна характеризується поступовим переходом від класичних, детермінованих алгоритмів розрахунку освітлення до імовірнісних методів генеративного штучного інтелекту. Традиційним промисловим стандартом фотореалістичної візуалізації сьогодні є метод трасування шляхів (Path Tracing), який базується на чисельному розв’язанні рівняння рендерингу [3]. Попри те, що фізична коректність цього підходу гарантує високий реалізм, його обчислювальна складність зростає експоненційно зі збільшенням кількості джерел світла та деталізації геометричної складної сцени. Для отримання зображення без шуму необхідно розрахувати тисячі світлових шляхів для кожного пікселя. Емпіричні дослідження продуктивності сучасних рендер-рушіїв свідчать, що візуалізація одного кадру у роздільній здатності 4K на споживчому обладнанні може тривати від кількох десятків хвилин до кількох годин. Окрім часових затрат, традиційний метод висуває жорсткі вимоги до апаратного забезпечення користувача, зокрема до обсягу відеопам’яті (VRAM). Спроба відрендерити сцену з використанням високоякісних PBR-матеріалів часто призводить до переповнення пам’яті графічного процесора та переходу в режим вивантаження даних в оперативну пам’ять (Out-of-core rendering), що додатково знижує швидкість роботи у десятки разів. Це створює нагальну потребу в розробці та впровадженні нових архітектурних рішень, здатних оптимізувати процес візуалізації.

Проаналізувати еволюцію інженерних підходів та існуючі архітектури генеративного штучного інтелекту в контексті їх застосування для оптимізації процесів 3D-візуалізації, а також обґрунтувати вибір оптимальної технології для подолання поточних апаратних обмежень користувачів.

Поняття нейронного рендерингу охоплює широкий спектр методів, що еволюціонували від простих інструментів пост-обробки до комплексних генеративних систем. Першим успішним етапом впровадження нейромереж у виробничі пайплайни стала технологія інтелектуального знешумлення (AI-Denoising). Алгоритми цього класу використовують згорткові автоенкодери та допоміжні геометричні буфери (карти кольору Albedo та карти нормалей Normal Pass) для відокремлення корисного візуального сигналу від стохастичного шуму [5]. Хоча цей метод дозволяє прискорити візуалізацію у кілька разів, він залишається суто інструментом реконструкції, який не здатний генерувати нові деталі чи оптимізувати роботу з важкими текстурами.

Наступним, значно потужнішим етапом стала поява латентних дифузійних моделей (Latent Diffusion Models, LDM), яскравими представниками яких є сімейство Stable Diffusion та розроблені на їх базі плагіни для 3D-редакторів. Фундаментальна ідея цього підходу полягає у перенесенні ітеративного процесу знешумлення з піксельного простору у стиснутий векторний (латентний) простір за допомогою варіаційного автоенкодера (VAE) [4]. Проте, при спробі інтеграції дифузійних моделей у професійні процеси 3D-візуалізації виявляється низка критичних архітектурних недоліків. Насамперед, такі системи створюють новий апаратний бар’єр: для стабільної генерації зображень потрібні відеокарти з обсягом пам’яті не менше 12–14 ГБ. У разі нестачі ресурсів система використовує повільну оперативну пам’ять, що нівелює будь-який виграш у швидкості. Крім того, традиційні LDM мають суттєві обмеження контексту через використання текстового енкодера CLIP, який обрізає вхідну послідовність після 77 токенів, унеможливлюючи точний опис складної сцени. Найбільш вразливим місцем таких архітектур є явище просторової агнозії (Spatial Agnosia). Оскільки модель навчалася переважно на двовимірних зображеннях, вона не має

глибинного розуміння тривимірних відношень, що призводить до некоректної інтерпретації взаємного розташування об’єктів та проблем із масштабуванням генерації до нативної роздільної здатності 4K.

Сучасним вирішенням цих проблем є перехід до архітектури мультимодальних великих мовних моделей (Multimodal LLM) на базі трансформерів, таких як Google Gemini. Фундаментальна відмінність цього підходу полягає у концепції уніфікованого сприйняття. Замість використання ізольованих потоків для тексту та зображень, мультимодальні трансформери розбивають вхідну візуальну інформацію (наприклад, карту глибини) на сітку фіксованих фрагментів (патчів). Кожен патч проходить через шар лінійної проєкції та отримує позиційне кодування, перетворюючись на послідовність візуальних токенів, які обробляються спільно з текстовими інструкціями механізмом глобальної само-уваги (Self-Attention) [6; 7].

Завдяки такій архітектурі система здобуває емерджентну властивість просторового інтелекту. Модель здатна виконувати складні логічні операції з геометрією, коректно інтерпретуючи градієнти яскравості на карті глибини як метричну інформацію про відстань. Це дозволяє генерувати складні оптичні ефекти, обчислювати явища оклюзії та вибудовувати логічні зв’язки освітлення ще до початку візуалізації (Visual Chain-of-Thought). Більше того, трансформери оперують послідовностями токенів довільної довжини, що знімає жорстку прив’язку до роздільної здатності і дозволяє здійснювати генерацію у нативному форматі 4K. Перенесення цих обчислювальних процесів у хмарне середовище повністю усуває проблему локального «вузького місця» відеопам’яті, демократизуючи доступ до високоякісного рендерингу [8].

Проведений аналіз архітектурних підходів демонструє, що традиційні методи візуалізації та локальні дифузійні моделі вичерпують свій потенціал через надмірні вимоги до апаратного забезпечення та нездатність до глибокого розуміння тривимірного простору. Інтеграція хмарних мультимодальних трансформерів у конвеєр 3D-моделювання є найбільш перспективним напрямком. Використання таких моделей дозволяє розгорнути клієнт-серверну

архітектуру, яка ефективно інтерпретує структурні дані сцени (карти глибини) та забезпечує генерацію фізично коректних зображень високої роздільної здатності, мінімізуючи навантаження на локальні обчислювальні ресурси користувача.

Список використаних джерел:

1. 3D Rendering Market Size, Share, and Trends 2025 to 2034. URL: https://www.precedenceresearch.com/3d-rendering-market (дата звернення: 12.03.2026).
2. 3D Rendering Market (2025–2033). URL: https://www.grandviewresearch.com/industry-analysis/3d-rendering-market-report (дата звернення: 12.03.2026).
3. Kajiya J. T. The rendering equation // *ACM SIGGRAPH Computer Graphics*. 1986. Vol. 20, No. 4. P. 143–150. DOI: https://doi.org/10.1145/15886.15902.
4. Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B. High-resolution image synthesis with latent diffusion models. arXiv. 2022. URL: https://arxiv.org/abs/2112.10752 (дата звернення: 12.03.2026).
5. Chaitanya C. R. A., Kaplanyan A. S., Schied C., Salvi M., Lefohn A., Nowrouzezahrai D., Aila T. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. ACM Transactions on Graphics. 2017. Vol. 36, No. 4. P. 1–12. DOI: https://doi.org/10.1145/3072959.3073601.
6. Dosovitskiy A. et al. An image is worth 16×16 words: transformers for image recognition at scale. arXiv. 2020. URL: https://arxiv.org/abs/2010.11929 (дата звернення: 12.03.2026).
7. Wu B. et al. Visual transformers: token-based image representation and processing for computer vision. arXiv. URL: https://arxiv.org/abs/2006.03677 (дата звернення: 12.03.2026).
8. Gemini Team. Gemini: a family of highly capable multimodal models. arXiv. 2023. URL: https://arxiv.org/abs/2312.11805 (дата звернення: 12.03.2026).

УДК 004.7:004.9

ВИКЛИКИ ТА РІШЕННЯ ІНЖЕНЕРІЇ МОБІЛЬНИХ КІБЕРФІЗИЧНИХ СИСТЕМ

Мазикін О.А.
oleksandrmazikin@gmail.com
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м. Черкаси, Україна

Традиційно інформаційні технології будувалися як ізольовані та стабільні системи. Класичний дата-центр або серверна – це добре контрольоване середовище з налаштованим кліматом, резервними джерелами живлення та чітко визначеним розташуванням обладнання. Уся інфраструктура заздалегідь спроєктована, а зміни вносяться рідко і за певними правилами. У таких умовах адміністрування є відносно простим: легко відстежити роботу систем, контролювати навантаження та швидко знаходити несправності. Якщо виникає проблема, її зазвичай можна локалізувати до конкретного сервера, мережевого пристрою або кабелю. Це значно спрощує обслуговування та ремонт.

Крім того, фізичний доступ до обладнання суворо обмежений. Використовуються системи безпеки, контроль доступу та фізичні бар'єри, що захищають інфраструктуру від несанкціонованого втручання. Завдяки цьому ризики, пов'язані з людським фактором або зовнішніми впливами, мінімізуються. Загалом така модель забезпечує високу надійність і передбачуваність роботи систем, але водночас вона є менш гнучкою і гірше пристосованою до швидких змін та масштабування.

Сьогодні IT-галузь переживає суттєві зміни: обчислювальні ресурси більше не зосереджені лише в дата-центрах, а розподіляються і працюють безпосередньо у фізичному середовищі. Наприклад, сучасні промислові дрони, складські AGV-системи (Automated Guided Vehicles) чи роботизовані платформи - це вже не просто техніка, а повноцінні мобільні обчислювальні вузли. Вони мають достатню потужність для обробки даних на місці, зокрема для роботи з

нейромережами, а також оснащені різними сенсорами (лідари, інерціальні системи) та виконавчими механізмами.

Ключова проблема мобільних кіберфізичних систем полягає в тому, як забезпечити стабільну та передбачувану передачу даних. На відміну від статичних серверів з дубльованою оптоволоконною лінією, рухомі агенти залежать від бездротових технологій, які піддаються впливу фізичних перешкод, радіоінтерференції та багатопроменевого поширення сигналу [1].

У масштабних індустріальних середовищах мобільні агенти змушені постійно перемикатися між точками доступу, а стандартний протокол Wi-Fi функціонує за парадигмою «break-before-make» (відключення перед новим з'єднанням). Це ініціює затримки під час хендоверу (handoff), що генерує втрату пакетів даних. Для системи, яка працює в режимі реального часу, затримка у кілька сотень мілісекунд може призвести до критичної просторової помилки.

Враховуючи ймовірність деградації сигналу, архітектура має передбачати автономні протоколи безпеки (Network Loss Autonomy):. При втраті з'єднання ініціюються апаратні сторожові таймери (Watchdogs). Відповідно до алгоритму, система або переходить у безпечний стан (Safe Mode) з плавною зупинкою, або продовжує місію, спираючись виключно на локальні обчислювальні потужності (Edge Computing) та бортову сенсорику[2].

Програмне забезпечення кіберфізичних систем кардинально відрізняється від класичних десктопних чи серверних рішень. Воно вимагає детермінованості, яку забезпечують операційні системи реального часу (RTOS), та модульного підходу до обробки даних.

Фреймворк ROS (Robot Operating System), незважаючи на назву, не є повноцінною операційною системою. Це спеціалізований middleware, який працює поверх UNIX-подібних систем, найчастіше Ubuntu.

ROS надає зручну архітектуру для побудови робототехнічних систем: окремі компоненти (наприклад, драйвери пристроїв або модулі навігації) працюють як незалежні вузли (nodes) і обмінюються даними через спеціальний

механізм «публікатор–підписник». Такий підхід дозволяє гнучко комбінувати різні частини системи та масштабувати її.

Водночас ефективність роботи всієї системи значною мірою залежить від правильної організації цього графа вузлів. Тому оптимізація взаємодії між вузлами, зменшення затримок і навантаження на мережу є важливим завданням для сучасного адміністратора або розробника [3].

Щоб забезпечити масштабованість і уникнути конфліктів між програмними компонентами, використовують контейнеризацію, наприклад, за допомогою Docker або його легших аналогів для вбудованих систем. Це дає змогу ізолювати окремі процеси разом із їхніми залежностями та забезпечити їхню стабільну роботу [4].

Еволюція обчислювальних систем не просто вдосконалює технології, а й кардинально змінює саму логіку побудови безпеки. Якщо класична кібербезпека фокусується на захисті конфіденційності та цілісності інформації, то кіберфізична безпека (Cyber-Physical Security) спрямована на захист фізичного середовища від скомпрометованих апаратних комплексів [2].

У сучасних умовах проблема вразливості обчислювальних систем набуває багатовимірного характеру, виходячи за межі суто інформаційних чи фінансових ризиків. Якщо компрометація бази даних традиційно асоціюється з економічними втратами, то несанкціонований доступ до промислових маніпуляторів, аграрних безпілотних апаратів або медичних роботизованих систем створює безпосередню загрозу життю та здоров’ю людей. Таким чином, безпека інформаційних технологій трансформується у сферу, де цифрові ризики мають прямі фізичні наслідки.

Особливу увагу в цьому контексті слід приділяти захисту апаратних інтерфейсів. Мобільні системи функціонують у відкритому середовищі, що підвищує ймовірність фізичного доступу зловмисників до портів вводу-виводу, зокрема USB та UART. З огляду на це сучасні протоколи безпеки передбачають комплекс превентивних заходів: відключення налагоджувальних інтерфейсів на рівні ядра операційної системи, застосування зашифрованих завантажувачів

(Secure Boot), а також апаратне блокування потенційних точок входу до системи. Такий підхід формує нову парадигму захисту, де інтеграція програмних і апаратних механізмів стає необхідною умовою забезпечення цілісності та стійкості обчислювальних систем.

Отже, інтеграція обчислювальних потужностей у мобільні фізичні об’єкти зумовлює необхідність переосмислення ролі фахівців з інфраструктури, адже їхня діяльність більше не обмежується суто програмним адмініструванням. У сучасних умовах інженер у сфері RoboOps постає як мультидисциплінарний експерт, здатний поєднувати компетенції мережевого архітектора, спеціаліста з кіберфізичної безпеки та системного інженера. Такий фахівець не лише володіє знаннями щодо побудови та захисту мережевих структур, але й розуміє фізику процесів, що відбуваються у взаємодії апаратних і програмних компонентів у реальному часі. Це забезпечує комплексний підхід до управління роботизованими системами, де технічна експертиза поєднується з глибоким розумінням алгоритмічних та фізичних аспектів їхньої роботи.

Список використаних джерел:

1. Quigley M. et al. ROS: an open-source Robot Operating System // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA Workshop). 2009. P. 1–6.
2. Lee E. A. Cyber-Physical Systems: Design Challenges // Proceedings of the 11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC). 2008. P. 363–369.
3. Merkel D. Docker: Lightweight Linux Containers for Consistent Development and Deployment // Linux Journal. 2014. № 239. P. 2.
4. Maruyama Y., Kato S., Azumi T. Exploring the Performance of ROS2 // Proceedings of the 13th International Conference on Embedded Software. – 2016. P. 1–10.

UDC 004.056:004.8

DETECTION AND CLASSIFICATION OF NETWORK INTRUSIONS USING ENSEMBLE LEARNING METHODS

Ivchenko. V.V.
valera2071w@gmail.com
Cherkasy State Business College
Ivanova. I.V.
Cherkasy, Ukraine

The presented work is dedicated to the development of intelligent systems for identifying cyber threats and network anomalies. This study was implemented within the framework of a specialized English-language Python programming course. A feature of the training was the full integration of technical English in the educational process: from studying the documentation of machine learning libraries to communicating with teachers and colleagues exclusively in English for discussing model architectures and experimental results.

The rapid development of cloud technologies and network infrastructure allows for the processing of vast amounts of traffic, which significantly increases the need for robust security monitoring in corporate and state systems. At the same time, the increasing complexity of cyber-attacks creates new challenges in the field of information processing and real-time pattern recognition.

In the conditions of work on the project «Network Intrusion Detection» based on the Kaggle platform, an analysis was conducted of data sequences representing network packet attributes. The main task was to predict different types of network behavior: normal traffic and various categories of malicious activities, including DoS attacks and unauthorized access attempts.

Among the main factors that provided high precision in the developed system, the following can be highlighted:

- Using the XGBoost gradient boosting algorithm, which demonstrated high efficiency in processing non-linear dependencies in network data.

- Thorough preliminary data processing, which included normalization of numerical features, encoding of categorical variables, and handling of imbalanced classes.
- Feature engineering aimed at identifying the most significant indicators of anomalous behavior in traffic flows.
- Application of comprehensive evaluation metrics (Precision, Recall, F1-score) for a deep analysis of model performance in security-critical conditions.

An important role in the implementation of the project was played by proficiency in technical English, which allowed for the effective use of global resources, such as the Kaggle platform, and the integration of modern machine learning techniques into the development. According to the results of testing, a high classification accuracy (0.94) was achieved, which confirms the efficiency of the chosen engineering approach.

In this way, it can be concluded that the combination of Python development skills with fluency in English in a professional environment is a key factor in the successful implementation of complex IT security projects. Effective threat detection requires a comprehensive approach that includes using modern machine learning technologies and constant improvement of a specialist's professional competencies.

References:

1. Kaggle. Network Intrusion Detection Dataset. Access mode: kaggle.com.
2. XGBoost Developers. XGBoost Documentation. Access mode: readthedocs.io.
3. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD, 2016. P. 785–794.
4. NIST Special Publication 800-94. Guide to Intrusion Detection and Prevention Systems. Gaithersburg, 2021.

УДК 004.8:004.056

ГЕНЕРАТИВНИЙ ШТУЧНИЙ ІНТЕЛЕКТ У СИСТЕМАХ ОБРОБКИ ТА ЗАХИСТУ ІНФОРМАЦІЙНИХ РЕСУРСІВ

Макаренко Д. В.
dimamakar188@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р. В.
м. Черкаси, Україна

Генеративний штучний інтелект стрімко розвивається і дедалі активніше використовується у сфері обробки та захисту інформаційних ресурсів. Сучасні алгоритми здатні не лише аналізувати великі масиви даних, але й генерувати новий контент, моделі поведінки, сценарії кіберзагроз або варіанти рішень для їх запобігання. Завдяки цьому генеративні моделі стають важливим інструментом у системах інформаційної безпеки.

Ідея автоматизованого аналізу інформації виникла задовго до появи сучасного генеративного штучного інтелекту. Традиційні системи кібербезпеки базувалися переважно на сигнатурному аналізі, тобто порівнянні підозрілих дій або файлів із заздалегідь відомими шаблонами загроз. Хоча цей підхід був ефективним для виявлення відомих атак, він мав суттєві обмеження, зокрема, нові або модифіковані загрози часто залишалися непоміченими. Поява машинного навчання дала можливість системам безпеки самостійно виявляти аномалії в мережевому трафіку та поведінці користувачів. Порівняння підходів до забезпечення інформаційної безпеки показано в табл.1.

Генеративний штучний інтелект став наступним етапом розвитку цих технологій. На відміну від класичних моделей машинного навчання, які здебільшого виконують класифікацію або прогнозування, генеративні моделі здатні створювати нові дані на основі вивчених закономірностей. У контексті інформаційної безпеки це дозволяє моделювати можливі сценарії атак, створювати синтетичні набори даних для навчання систем захисту та тестувати стійкість інформаційних систем до нових типів загроз.

Технічною основою генеративного штучного інтелекту є глибокі нейронні мережі, які навчаються на великих масивах даних. Серед найбільш поширених підходів – генеративні змагальні мережі, Variational Autoencoder та трансформерні моделі. Генеративні змагальні мережі працюють за принципом змагання двох нейромереж, у яких одна генерує дані, а інша оцінює їхню достовірність.

Одним із важливих напрямів використання генеративного штучного інтелекту є виявлення аномалій. Наприклад, система може виявити незвичайну активність у мережі, спробу несанкціонованого доступу або аномальні зміни у файлах. Крім того, аналізуючи великі обсяги даних, такі системи можуть виявляти можливі загрози та пропонувати варіанти їх усунення [1].

Дослідники можуть застосовувати генеративний штучний інтелект для створення синтетичних даних, що допомагає безпечно вивчати поведінку шкідливого програмного забезпечення та вдосконалювати біометричні системи захисту [2]. Багато організацій вже використовують штучний інтелект для аналізу загроз і оцінки вразливостей [3].

Зловмисники можуть застосовувати такі технології для створення фішингових повідомлень, підроблених документів, deepfake-контенту або автоматизованих атак. Генеративні моделі здатні формувати переконливі тексти та імітувати стиль реальних людей, що підвищує ефективність соціальної інженерії. Зокрема, попри зростання рівня автоматизації, системи безпеки потребують постійного контролю з боку фахівців, щоб уникнути помилок і зловживань [4].

Однією з ключових проблем є здатність генеративних моделей створювати неточну або оманливу інформацію , що може призвести до неправильного аналізу загроз і прийняття хибних рішень [5].

Системи на основі генеративного штучного інтелекту можуть не лише виявляти загрозу, але й пропонувати або навіть автоматично виконувати заходи з її нейтралізації. Наприклад, система може ізолювати заражений вузол мережі, заблокувати підозрілу активність або змінити політику доступу до ресурсів.

Використання генеративного штучного інтелекту також пов’язане з етичними та правовими питаннями. Необхідно забезпечити прозорість роботи алгоритмів, захист персональних даних та відповідальність за прийняті автоматизованими системами рішення.

Таблиця 1 – Порівняння підходів до забезпечення інформаційної безпеки

№	Характеристика	Традиційні системи безпеки	Системи на основі машинного навчання	Системи на основі генеративного штучного інтелекту
1	Принцип роботи	Використання сигнатур і правил для виявлення відомих загроз	Аналіз поведінки та статистичних закономірностей у даних	Генерація нових сценаріїв атак та моделей поведінки
2	Тип загроз, що виявляються	Переважно відомі загрози	Відомі та деякі нові аномалії	Відомі, нові та потенційні майбутні сценарії атак
3	Необхідність оновлення	Часте оновлення сигнатур	Потрібне перенавчання моделей	Самонавчання та генерація нових даних для навчання
4	Робота з даними	Порівняння з базою шаблонів	Аналіз великих масивів даних	Генерація синтетичних даних та моделювання загроз
5	Автоматизація	Низький рівень	Середній рівень	Високий рівень автоматизації та адаптації
6	Основні переваги	Простота та швидкість виявлення відомих атак	Виявлення аномалій	Прогнозування та моделювання нових атак
7	Основні недоліки	Не виявляє нові загрози	Потребує великих наборів даних	Можливі помилки генерації та ризик зловживань

У результаті проведеного дослідження встановлено, що генеративний штучний інтелект є перспективним інструментом у системах обробки та захисту інформаційних ресурсів. На відміну від традиційних підходів, він дозволяє не лише виявляти відомі загрози, а й моделювати нові сценарії атак, виявляти аномалії. Разом із тим використання генеративного штучного інтелекту пов’язане з певними ризиками, зокрема можливістю генерації неточної інформації та використанням цих технологій зловмисниками. Тому його

впровадження потребує поєднання автоматизованих рішень із контролем з боку фахівців.

Список використаних джерел:

1. Generative AI in Cybersecurity: Balancing Innovation and Risk. URL: https://cert.europa.eu/publications/security-guidance/generative-ai-in-cybersecurity-balancing-innovation-and-risk/ (дата звернення: 13.03.2026).
2. What Is Generative AI in Cybersecurity? URL: https://www.paloaltonetworks.com/cyberpedia/generative-ai-in-cybersecurity (дата звернення: 14.03.2026).
3. What Is Generative AI in Cybersecurity? URL: https://www.sentinelone.com/cybersecurity-101/data-and-ai/generative-ai-cybersecurity/ (дата звернення: 13.03.2026).
4. How Can Generative AI Be Used In Cybersecurity? (Ultimate Guide) URL: https://www.eweek.com/artificial-intelligence/generative-ai-and-cybersecurity/ (дата звернення: 14.03.2026).
5. Generative AI in cybersecurity. URL: https://www.capgemini.com/insights/research-library/generative-ai-in-cybersecurity/ (дата звернення: 13.03.2026).

УДК 004.8

ВИКОРИСТАННЯ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ СТВОРЕННЯ ДИНАМІЧНИХ НАРАТИВІВ ТА ДІАЛОГІВ У ВІДЕОІГРАХ

Шкода В.В.
vlad2009skoda@gmail.com
Черкаський державний фаховий бізнес-коледж
Литовченко В.О.
м. Черкаси, Україна

Сучасна індустрія відеоігор активно розвивається та прагне створювати більш реалістичні та цікаві ігрові світи. Важливу роль у цьому відіграє взаємодія гравця з неігровими персонажами (NPC) та можливість впливати на розвиток сюжету [2].

У традиційних іграх діалоги та сюжет створюються заздалегідь. Наприклад, у грі Detroit: Become Human використовується складна система розгалужених рішень, де події змінюються залежно від вибору гравця (рис. 1). Такий підхід дозволяє створити цікаву історію, але потребує багато часу на розробку, оскільки всі варіанти потрібно прописувати вручну.

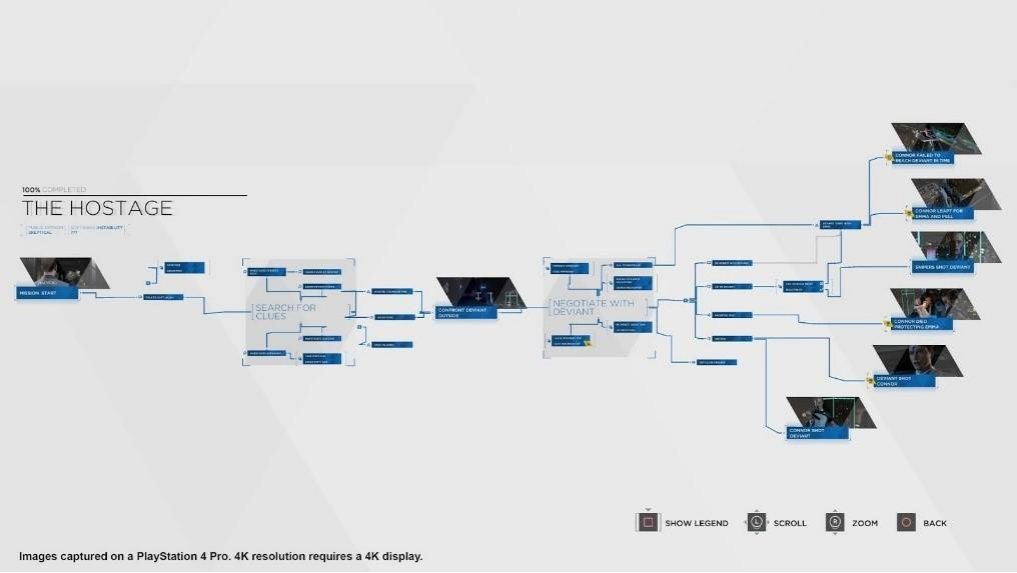


Рисунок 1 – Приклад розгалуженої системи сюжетних рішень у відеогрі Detroit: Become Human

Сучасні технології дають можливість використовувати генеративний штучний інтелект, зокрема великі мовні моделі (Large Language Models, LLM),

які можуть створювати діалоги прямо під час гри [3]. Завдяки цьому персонажі можуть реагувати на дії гравця більш гнучко та природно.

У сучасних відеоіграх все частіше використовуються такі підходи, тому цікаво розглянути, як саме генеративний штучний інтелект може застосовуватися для створення діалогів і сюжетів.

Зазвичай впровадження генеративного штучного інтелекту в ігрові системи включає кілька основних етапів (табл. 1): створення контексту для персонажів, аналіз дій гравця та генерація відповідей у реальному часі [2].

Використання генеративного штучного інтелекту дозволяє зробити ігровий процес більш різноманітним. Кожне проходження може відрізнятися, оскільки діалоги не повторюються. Це також зменшує навантаження на розробників, адже частину контенту можна створювати автоматично [2].

Таблиця 1 – Етапи впровадження генеративного штучного інтелекту в ігрові системи

№	Етап впровадження	Що це дає розробці	Ефект для гравця
1	Контекстне моделювання	Формування бази знань і характеру персонажа	Персонажі поводяться відповідно до своєї ролі
2	Аналіз дій гравця	Врахування вибору та стилю гри у реальному часі	Світ реагує на дії гравця
3	Генерація діалогів	Автоматичне створення реплік	Кожне проходження має унікальні діалоги
4	Синтез мовлення	Озвучення згенерованого тексту	Діалоги звучать природно

Разом з тим існують і певні труднощі. Наприклад, іноді система може генерувати нелогічні або некоректні відповіді, що може порушити цілісність сюжету. Також важливими є питання авторського права та використання голосів акторів [4].

Отже, генеративний штучний інтелект відкриває нові можливості для розвитку відеоігор. Його використання дозволяє створювати більш динамічні та інтерактивні історії.

Список використаних джерел:

1. NVIDIA. NVIDIA ACE for Games: Bringing Digital Humans to Life. 2024. URL: https://developer.nvidia.com/ace-for-games (дата звернення: 19.03.2026).
2. Unity. Generative AI in Game Development: How It’s Changing the Industry . – 2024. – URL: https://blog.unity.com/engine-platform/generative-ai-in-game-development (дата звернення: 19.03.2026).
3. Inworld AI. The Future of NPCs: How Inworld AI is Changing Gaming. 2025. URL: https://inworld.ai/blog/the-future-of-npcs (дата звернення: 19.03.2026).
4. Game Developer. AI-Powered Storytelling: The Next Frontier in Game Narrative. 2024. URL: https://www.gamedeveloper.com/design/ai-powered-storytelling (дата звернення: 19.03.2026).

УДК 004.8:004.49

ВПЛИВ ШІ НА ЕВОЛЮЦІЮ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА АНТИВІРУСІВ

Федоров Є.О.
egorfedorov0306@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В.Ю.
м. Черкаси, Україна

Стрімке впровадження технологій штучного інтелекту та машинного навчання кардинально змінило глобальний ландшафт кібербезпеки. Сучасні інформаційні системи стикаються з викликами, які вимагають принципово нових підходів до захисту даних. Якщо раніше протистояння між зловмисниками та антивірусними компаніями базувалося на статичному аналізі файлів та сигнатурах, то сьогодні ця боротьба перетворилася на високотехнологічну алгоритмічну війну. Штучний інтелект став одночасно і найпотужнішим щитом, і найнебезпечнішою зброєю.

Використання машинного навчання відкрило нові вектори атак, головним з яких є створення інтелектуального поліморфного коду. Сучасне шкідливе ПЗ

здатне динамічно переписувати власну структуру безпосередньо під час виконання.

Зберігаючи свій первинний деструктивний функціонал, такий вірус після кожного нового зараження генерує унікальну цифрову сигнатуру. Це робить кожен його екземпляр абсолютно унікальним, зводячи нанівець ефективність класичних антивірусних сканерів.

Крім того, кіберзлочинці активно застосовують ШІ для автоматизації пошуку вразливостей нульового дня. Спеціально навчені нейромережі здатні за лічені години проаналізувати мільйони рядків коду операційних систем, знаходячи приховані архітектурні хиби, на пошук яких у людей пішли б місяці.

Ще одним критичним досягненням "темного" ШІ стало "розумне ухилення" (AI-driven Evasion). Інтелектуальні віруси навчилися самостійно аналізувати середовище свого виконання. Якщо такий алгоритм розпізнає ознаки віртуальної пісочниці (Sandbox) або фіксує активний моніторинг, він миттєво призупиняє шкідливу активність. Вірус починає ідеально імітувати поведінку звичайного системного процесу, доки не переконається у власній безпеці.

Розглядаючи еволюцію кіберзагроз, неможливо оминути ще один інноваційний та вкрай небезпечний вектор – цілеспрямовані атаки на самі моделі машинного навчання, які лежать в основі сучасних антивірусів. Цей науковий напрямок отримав назву змагального машинного навчання (Adversarial Machine Learning). Зловмисники усвідомили: замість того, щоб намагатися непомітно обійти "розумний" захист, набагато ефективніше змусити помилятися на фундаментальному математичному рівні.

Однією з тактик є "отруєння даних". Її суть полягає в тому, що хакери тривалий час і дуже обережно "згодовують" системі захисту EDR (Endpoint Detection and Response) спеціально модифіковані, зовні безпечні зразки активності. Головна мета цього процесу – змусити нейромережу змінити своє уявлення про "нормальну поведінку" системи. Якщо атакувальнику вдасться поступово змістити базову лінію норми у математичній моделі, штучний

інтелект почне ігнорувати реальні шкідливі процеси, помилково класифікуючи їх як легітимні дії користувача.

Іншим передовим методом є використання змагального шуму (Adversarial Perturbations). Зловмисники вносять у код шкідливої програми мінімальні, функціонально незначущі зміни – своєрідний невидимий "цифровий шум", який абсолютно не впливає на роботу вірусу. Проте для математичної моделі антивірусу цей шум кардинально спотворює сприйняття об'єкта. Через такі маніпуляції нейромережа стикається з оптичною ілюзією на рівні коду і може впевнено класифікувати небезпечний руткіт як безпечний системний драйвер або текстовий документ. Це створює парадоксальну ситуацію в сучасній кібербезпеці: чим складніший і "чутливіший" штучний інтелект захисної системи, тим вразливішим він може виявитися до таких специфічних математичних маніпуляцій.

У відповідь на ці безпрецедентні виклики індустрія безпеки здійснила перехід від реактивного до проактивного захисту. Сучасні системи класу EDR більше не покладаються виключно на бази відомих загроз. Ядром новітніх антивірусів стали ансамблі моделей машинного навчання, які здійснюють глибокий поведінковий аналіз у режимі реального часу. Такі системи безперервно збирають телеметрію, відстежуючи звернення програм до пам'яті та мережі.

На основі зібраних даних штучний інтелект формує унікальний профіль нормальної поведінки для кожної машини. Будь-яке відхилення від цієї норми (наприклад, спроба масового шифрування файлів) миттєво класифікується як аномалія. Це дозволяє системі захисту приймати автономні рішення. Вона може автоматично заблокувати підозрілий процес та ізолювати пристрій від загальної мережі ще до того, як комп'ютеру буде завдано фактичної шкоди.

Окремою і надзвичайно важливою тенденцією є взаємне використання генеративно-змагальних мереж (GAN) обома сторонами конфлікту. Зловмисники застосовують GAN для генерації мільйонів варіацій шкідливого коду, тренуючи їх обходити комерційні антивіруси. Своєю чергою, розробники

систем захисту використовують ці ж технології у своїх лабораторіях для створення "штучних вірусів". На цих згенерованих зразках вони превентивно тренують свої моделі виявлення, навчаючи їх розпізнавати загрози майбутнього.

Таким чином штучний інтелект назавжди змінив фундаментальні правила інформаційної безпеки. Відмова від класичних парадигм на користь машинного навчання стала єдиним шляхом виживання в сучасному цифровому світі.

Ефективність сучасного захисту сьогодні вимірюється не розміром вірусних баз, а обчислювальною потужністю його математичних моделей. У найближчому майбутньому кіберпростір остаточно перетвориться на арену автономних алгоритмічних битв, де участь людини буде зведена до контролю високорівневих політик безпеки.

Але незважаючи на революційний вплив штучного інтелекту, його впровадження не є абсолютною панацеєю від усіх сучасних кіберзагроз.

Список використаних джерел:

1. Звіт про глобальні кіберзагрози (Global Threat Report 2024). CrowdStrike. URL: https://www.crowdstrike.com/global-threat-report/ (Дата звернення: 07.04.2026)
2. Звіт про цифровий захист (Microsoft Digital Defense Report 2024). Microsoft. URL: https://www.microsoft.com/en-us/security/security-insider/threat-landscape/microsoft-digital-defense-report-2024 (Дата звернення: 07.04.2026)
3. AI в кібербезпеці: як штучний інтелект захищає і атакує цифровий світ. *Maxnet Blog*. URL: https://maxnet.ua/blog/ai-v-kiberbezpeci-yak-shtuchnij-intelekt-zahishaye-i-atakuye-cifrovij-svit/ (Дата звернення: 07.04.2026)
4. ENISA. Artificial Intelligence Cybersecurity Challenges. European Union Agency for Cybersecurity, 2020. URL: https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges (Дата звернення: 07.04.2026)
5. Gallagher S., Posey B. AI and the Future of Cybersecurity: Defensive and Offensive Applications. O'Reilly Media, 2022. 210 p.

6. IBM Security. *Cost of a Data Breach Report 2023* URL: https://www.ibm.com/security/data-breach (Дата звернення: 07.04.2026)

УДК 004.738.5

ПРОГНОЗУВАННЯ ПСИХОЛОГІЧНОГО СТАНУ КОРИСТУВАЧА НА ОСНОВІ ЦИФРОВИХ СЛІДІВ

Рибалко А. Д.
anna10rybalko@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

У цифрову епоху кожна дія людини в інформаційному просторі залишає «цифрові сліди» – сукупність даних про онлайн-активність, які відкривають нові можливості для аналізу ментального здоров'я. Прогнозування психологічного стану на основі цих даних є актуальним інструментом для превентивної психології та персоналізованої підтримки користувача [1].

Цифрові сліди включають широкий спектр інформації: тексти публікацій та коментарів у соціальних мережах, історію пошукових запитів, вподобання контенту, часові інтервали активності та навіть метадані фотографій [2]. Аналіз семантики повідомлень дозволяє виявляти ранні ознаки депресивних станів, тривожності або професійного вигорання.

Технологічною основою такого прогнозування є моделі обробки природної мови (NLP) та нейронні мережі [3]. Алгоритми здатні обробляти неструктуровані дані, розпізнавати складні патерни поведінки та зіставляти їх із психологічними маркерами.

Це дозволяє створювати динамічний профіль стану користувача, який оновлюється в режимі реального часу, на відміну від традиційних опитувальників [4].

Наприклад, зміна кола інтересів, часте відвідування ресурсів із песимістичним контентом або зміна нічного режиму активності можуть свідчити про порушення психологічної рівноваги [5]. Водночас аналіз цифрових слідів

дозволяє фіксувати й позитивну динаміку: зростання соціальної взаємодії, конструктивну активність у професійних спільнотах та стабільність інтересів.

Інтеграція таких систем у повсякденні сервіси дозволяє реалізувати функцію «психологічного асистента» [6]. Програма може вчасно порадити користувачеві зробити перерву, звернути увагу на рівень стресу або запропонувати техніки релаксації.

Основою функціонування подібних програм є методи штучного інтелекту та машинного навчання [7]. Алгоритми аналізують великі обсяги даних, виявляють приховані закономірності та формують індивідуальні моделі поведінки користувача.

Наприклад, різке зниження активності, уникнення спілкування, скорочення кількості повідомлень або зміни у швидкості друку можуть свідчити про стрес, втому чи емоційне виснаження. У свою чергу, підвищена активність може вказувати як на позитивні емоції, так і на тривожність залежно від контексту.

Важливим компонентом таких систем є модуль рекомендацій, який формує персоналізовані поради щодо покращення емоційного стану користувача. Однією з ключових переваг таких програм є можливість раннього виявлення негативних емоційних станів, що дозволяє своєчасно реагувати та запобігати їх розвитку. Разом із тим використання подібних технологій пов’язане з низкою викликів, серед яких особливе місце займають питання конфіденційності та безпеки даних [8]. Оскільки обробляється персональна інформація, необхідно забезпечити її захист.

Крім того, важливо враховувати етичні аспекти використання таких систем, а також застосовувати сучасні підходи до кібербезпеки та захисту інформації [9].

Таким чином, прогнозування психологічного стану на основі цифрових слідів є перспективним напрямком розвитку інформаційних технологій, що поєднує аналіз великих даних, штучний інтелект та турботу про психічне здоров’я людини. Подальший розвиток таких систем сприятиме підвищенню

якості життя користувачів та формуванню більш усвідомленого ставлення до власного емоційного стану.

Список використаних джерел:

1. Harari G. M., Lane N. D., Wang R. et al. Using Smartphones to Collect Behavioral Data in Psychological Science // *Nature Human Behaviour*. 2016.
2. Miller G. The Smartphone Psychology Manifesto // *Perspectives on Psychological Science*. 2012.
3. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. MIT Press, 2016.
4. Wang R. et al. StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones // *Proceedings of ACM UbiComp*. 2014.
5. Saeb S. et al. Mobile Phone Sensor Correlates of Depressive Symptom Severity in Daily-Life Behavior // *Journal of Medical Internet Research*. 2015.
6. Lane N. D., Georgiev P., Qendro L. DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments // *Proceedings of ACM UbiComp*. 2015.
7. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson, 2021.
8. GDPR Portal. General Data Protection Regulation (GDPR). URL: https://gdpr.eu/ (дата звернення: 27.03.2026).
9. Microsoft. Data Privacy and Security in AI Systems. URL: https://www.microsoft.com/security (дата звернення: 27.03.2026).

УДК 004.8:004.056

ШІ-АГЕНТИ В КІБЕРПРОСТОРІ: ЗАСТОСУВАННЯ В АТАКАХ ТА СИСТЕМАХ ЗАХИСТУ ІНФОРМАЦІЇ

Мотайленко О.О.
omotaylenko@gmail.com
Черкаський державний фаховий бізнес-коледж
Захарова М.В.
м. Черкаси, Україна

У сучасному цифровому суспільстві кіберпростір став одним із ключових середовищ функціонування інформаційних систем, цифрових сервісів та комунікацій. Швидкий розвиток інформаційних технологій, хмарних обчислень, Інтернету речей (IoT) та глобальних мереж сприяє значному зростанню обсягів даних і кількості підключених пристроїв, а разом із цим збільшується і кількість кіберзагроз, що створює необхідність у використанні нових технологій для забезпечення ефективного захисту інформації. Одним із перспективних напрямів розвитку кібербезпеки є використання систем штучного інтелекту та автономних ШІ-агентів, здатних аналізувати інформацію, приймати рішення та виконувати складні завдання без постійного контролю людини [1–3].

Метою даної роботи є дослідження ролі та можливостей ШІ-агентів у сучасному кіберпросторі, аналіз особливостей їх застосування як у процесі здійснення кібератак, так і в системах захисту інформації. У роботі передбачено розглянути сутність ШІ-агентів, основні напрями їх використання в кібербезпеці, переваги та ризики впровадження, а також перспективи розвитку цих технологій у сфері кіберзахисту з урахуванням технічних, етичних і правових аспектів.

ШІ-агенти - це програмні системи, які можуть автономно взаємодіяти з цифровим середовищем, збирати дані, аналізувати їх та виконувати певні дії відповідно до заданих алгоритмів або моделей машинного навчання. Завдяки використанню методів штучного інтелекту, таких як машинне навчання, глибоке навчання та аналіз великих даних, ці системи здатні обробляти величезні масиви інформації та знаходити закономірності, які складно виявити традиційними методами. Як показано на рис. 1, у сфері кібербезпеки ШІ-агенти можуть

виконувати функції моніторингу мережевого трафіку, аналізу поведінки користувачів, виявлення аномалій та автоматичного реагування на інциденти безпеки [2–4].

З іншого боку, ШІ-агенти активно використовуються для підвищення рівня кіберзахисту. Сучасні системи інформаційної безпеки дедалі частіше застосовують алгоритми машинного навчання для аналізу мережевого трафіку та виявлення підозрілої активності. Наприклад, системи виявлення вторгнень (IDS) та системи запобігання вторгненням (IPS) використовують моделі штучного інтелекту для аналізу поведінкових моделей користувачів і пристроїв у мережі. Якщо система виявляє відхилення від нормальної поведінки, вона може автоматично сигналізувати про потенційну атаку або навіть заблокувати підозрілу активність [1–4].

Рисунок 1 – Схема застосування ШІ-агентів у кіберпросторі

Особливо важливим напрямом є створення автономних систем кіберзахисту, здатних самостійно реагувати на загрози. Такі системи можуть автоматично блокувати шкідливі IP-адреси, ізолювати заражені пристрої у мережі або аналізувати нові типи шкідливого програмного забезпечення. Деякі

сучасні платформи кібербезпеки використовують концепцію self-healing systems, тобто систем, здатних самостійно відновлювати нормальну роботу після атаки та адаптувати механізми захисту до нових загроз. Це дозволяє значно скоротити час реагування на інциденти та зменшити навантаження на фахівців з кібербезпеки [2; 6; 7].

Таблиця 1 – Автоматизація різних видів кібератак

Вид кібератаки	Як відбувається автоматизація	Що це дає зловмисникам	Ризики для систем захисту
Фішинг	ШІ автоматично створює велику кількість правдоподібних повідомлень і підлаштовує їх під різні категорії користувачів	Підвищення масовості атак, економія часу, більша переконливість повідомлень	Користувачам важче відрізнити підроблені листи від справжніх
Соціальна інженерія	Алгоритми аналізують відкриті дані про людину та формують персоналізовані сценарії впливу	Атаки стають більш адресними та психологічно ефективними	Зростає ймовірність успішного обману користувачів
Пошук вразливостей	Автоматизовані системи швидко аналізують програми, мережі й сервіси на наявність слабких місць	Прискорення виявлення потенційних точок проникнення	Скорочується час між появою вразливості та її використанням
Шкідливе програмне забезпечення	ШІ може змінювати поведінку шкідливого коду для обходу стандартних засобів виявлення	Підвищення стійкості шкідливих програм до захисту	Традиційні сигнатурні методи виявлення працюють гірше
DDoS-атаки	Бот-мережі автоматично координують великий потік запитів до сервісів чи сайтів	Масштабність і безперервність перевантаження ресурсів	Сервери та мережеві служби можуть втрачати доступність
Підбір облікових даних	Автоматизовані механізми масово перевіряють викрадені або типові дані входу	Прискорення несанкціонованого доступу до акаунтів	Зростає кількість атак на автентифікацію
Дезінформаційні кампанії	Генеративні моделі швидко створюють тексти, зображення чи повідомлення для масового поширення	Масове охоплення та швидке поширення неправдивої інформації	Ускладнюється перевірка достовірності контенту
Адаптивні атаки	Алгоритми аналізують реакцію системи захисту та змінюють модель поведінки атаки	Підвищення шансів обійти захисні механізми	Системи безпеки повинні постійно адаптуватися до нових сценаріїв

Інтеграція штучного інтелекту з іншими технологіями, такими як блокчейн, хмарні обчислення та Інтернет речей, відкриває нові можливості для створення комплексних систем інформаційної безпеки. Наприклад, поєднання технологій блокчейну та штучного інтелекту може забезпечити більш надійний контроль доступу до даних, а використання ШІ в IoT-мережах дозволяє виявляти аномалії у роботі підключених пристроїв. Однак розширення цифрової інфраструктури також збільшує поверхню потенційних атак, що потребує постійного вдосконалення систем захисту [3; 9].

Разом із перевагами використання ШІ-агентів виникають і нові виклики. Одним із них є проблема прозорості алгоритмів штучного інтелекту, оскільки складні нейронні мережі можуть приймати рішення, які складно пояснити або перевірити. Крім того, існує ризик використання таких технологій у злочинних цілях, що може призвести до появи нових форм кібератак. У зв'язку з цим важливим є розвиток правових механізмів регулювання використання штучного інтелекту у сфері кібербезпеки та створення етичних стандартів його застосування [3; 8; 9].

Таким чином, ШІ-агенти стають важливим елементом сучасного кіберпростору. Вони можуть використовуватися як для здійснення складних кібератак, так і для ефективного захисту інформаційних систем. Подальший розвиток цих технологій потребує комплексного підходу, що поєднує технічні інновації, міжнародні стандарти безпеки та відповідальне використання штучного інтелекту. Лише за умови поєднання наукових досліджень, технологічного розвитку та правового регулювання можливо забезпечити ефективний захист інформації в умовах стрімкого розвитку цифрового середовища.

Список використаних джерел:

1. Microsoft. What is AI for cybersecurity? URL: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-ai-for-cybersecurity (дата звернення: 17.03.2026).

2. IBM. Artificial Intelligence in Cybersecurity. URL: https://www.ibm.com/topics/artificial-intelligence-cybersecurity (дата звернення: 16.03.2026).
3. ENISA. Artificial Intelligence and Cybersecurity: Opportunities and Challenges. URL: https://www.enisa.europa.eu/publications/artificial-intelligence-and-cybersecurity (дата звернення: 18.03.2026).
4. Cisco. AI and Cybersecurity: How AI is Transforming Security. URL: https://www.cisco.com/c/en/us/products/security/ai-cybersecurity.html (дата звернення: 15.03.2026).
5. CrowdStrike. Artificial Intelligence in Cybersecurity. URL: https://www.crowdstrike.com/cybersecurity-101/artificial-intelligence/ (дата звернення: 17.03.2026).
6. Palo Alto Networks. The Role of AI in Cybersecurity. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-ai-in-cybersecurity (дата звернення: 16.03.2026).
7. Fortinet. AI in Cybersecurity Explained. URL: https://www.fortinet.com/resources/cyberglossary/artificial-intelligence-in-cybersecurity (дата звернення: 18.03.2026).
8. Check Point. AI and Cybersecurity: Benefits and Risks. URL: https://www.checkpoint.com/cyber-hub/cyber-security/what-is-ai-in-cybersecurity/ (дата звернення: 15.03.2026).
9. Cloudflare. AI Security: Threats and Protection. URL: https://www.cloudflare.com/learning/security/what-is-ai-security/ (дата звернення: 19.03.2026).

УДК 004.7:004.9

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ ПРОТОКОЛІВ MQTT ТА HTTP У IOT-СИСТЕМАХ

Сайко В.В.
vsa.st01@outlook.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Сучасний етап розвитку інформаційних технологій характеризується стрімкою цифровізацією та впровадженням систем Інтернету речей (IoT) у промисловість, енергетику та побутову сферу. Як показують дослідження сучасних тенденцій у галузі інформаційних технологій, ефективність таких систем критично залежить від стабільності обміну даними між кінцевими вузлами та центральними серверами. Вузли моніторингу часто будуються на базі малопотужних мікроконтролерів (ESP32 чи STM32), тож вибір мережевого протоколу стає визначальним фактором їхньої надійності та автономності.

Протокол HTTP є найпопулярнішим методом передачі даних у веб-середовищі. Але його значний недолік – високі накладні витрати. Кожен запит вимагає встановлення нового TCP з'єднання або підтримки механізму сесій, а обсяг текстових заголовків запитів може складати до 800 байтів навіть при передачі одного числового значення [1]. В мережах з сотнями пристроїв це призводить до навантаження на канали зв'язку, що зменшує автономність роботи пристроїв.

Альтернативнм рішенням є протокол MQTT (Message Queuing Telemetry Transport), який з самого початку розроблявся для умов з низькою пропускною здатністю. Замість моделі «запит-відповідь» в HTTP, MQTT використовує модель «видавець-підписник». Взаємодія проходить через брокер – центральний вузол, який керує потоками повідомлень. Така модель дозволяє пристроям більшу частину часу перебувати в режимі очікування, це дуже важливо для енергоефективності [2]. Одна з найбільших переваг MQTT – його бінарна структура. Заголовок пакета має розмір 2 байти, що в рази менше, ніж у

звичайного HTTP-заголовка. Порівняння показує, що при відправці повідомлення обсягом 10 байтів, загальний пакет даних у MQTT буде в 10–12 разів меншим, ніж у HTTP.

Для стабільного моніторингу в реальних умовах MQTT покладається на QoS (Quality of Service). Рівень «QoS 1» гарантує доставку повідомлення хоча б один раз, що важливо для точного збору телеметрії. Важливим моментом є контроль технічного стану самих IoT-пристроїв. Тут MQTT пропонує свій механізм «Last Will and Testament». Коли пристрій раптово втрачає зв'язок через розряд батареї або технічний збій, брокер розсилає повідомлення про його відключення всім учасникам мережі. Щоб реалізувати такий функціонал в протоколі HTTP потрібно постійно опитувати пристрій (polling), створюючи надлишковий трафік та неефективні витрати енергії [3].

Ще одним важливим пунктом є безпека передачі даних. В HTTP, кожен запит може виконувати ресурсоємну процедуру TLS-рукостискання (handshake), в свою чергу MQTT встановлює зашифроване з’єднання один раз і підтримує його тривалий час. Протокол MQTT значну перевагу з архітектурою «Zero Trust» (нульової довіри). Брокер реалізує чітке розмежування мережі, де кожен датчик має права доступу лише до відповідних топіків, що робить неможливим подальше просування зловмисника у разі компрометації одного з вузлів. HTTP передає складні токени авторизації у кожному заголовку, що збільшує обсяг трафіку, навантаження на пристрій та вразливість до атак типу DoS через обмеженість ресурсів.

Отже, протокол MQTT є набагато ефективнішим за HTTP для систем IoT, яким потрібна висока стабільність та енергоефективність. Його використання зменшує обсяг трафіку на 80-90%, забезпечує гарантовану і захищену доставку повідомлень, збільшує автономність та дає можливість контролювати стан пристроїв у реальному часі.

Список використаних джерел:

1. RFC 7230: hypertext transfer protocol (HTTP/1.1): message syntax and routing. IETF Datatracker. URL: https://datatracker.ietf.org/doc/html/rfc7230 (дата звернення: 24.03.2026).
2. MQTT version 3.1.1. OASIS standard. URL: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html (дата звернення: 24.03.2026).
3. IoT fundamentals: networking technologies, protocols, and use cases for the internet of things. Cisco Press: Source for Cisco Technology. URL: https://www.ciscopress.com/store/iot-fundamentals-networking-technologies-protocols-9781587144561 (дата звернення: 24.03.2026).

УДК 004.8:336.7

ВИЯВЛЕННЯ ФІНАНСОВОГО ШАХРАЙСТВА У ЦИФРОВИХ ПЛАТІЖНИХ СИСТЕМАХ ІЗ ВИКОРИСТАННЯМ СУЧАСНИХ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Печерський Є.В.
yegor.141231@gmail.com
Черкаський державний фаховий бізнес-коледж
Марченко С. В.
м.Черкаси, Україна

Інтенсивна цифровізація фінансових сервісів, розвиток електронної комерції та мобільних платіжних платформ спричинили суттєве збільшення обсягів транзакційних потоків і водночас ускладнення шахрайських схем. Сучасні фінансові екосистеми характеризуються високою динамікою поведінкових патернів користувачів, мережевою природою взаємодій та значною нерівномірністю розподілу класів. За таких умов традиційні підходи на основі правил та класичні статистичні методи демонструють обмежену здатність до адаптації й своєчасного виявлення нових типів фінансових загроз.

Останні дослідження показують, що ефективність протидії шахрайству визначається не лише точністю класифікації транзакцій, але й здатністю систем аналізувати часову еволюцію фінансових взаємодій, враховувати мережеву

структуру даних і підтримувати стабільність роботи моделей у production-середовищі [1; 2]. У цьому контексті особливої актуальності набуває використання сучасних методів машинного навчання та штучного інтелекту, орієнтованих на обробку потокових і високо взаємопов'язаних даних.

Аналіз сучасних підходів машинного навчання до виявлення фінансового шахрайства, визначення їхніх функціональних переваг і практичних обмежень, а також обґрунтування архітектурних принципів побудови інтелектуальних систем аналізу транзакційних потоків у реальному часі.

Ранні системи виявлення фінансового шахрайства базувалися на алгоритмах класифікації із ручною інженерією ознак. Методи логістичної регресії, дерев рішень і ансамблеві алгоритми забезпечували прийнятні результати в умовах відносно стабільних даних. Однак такі моделі виявилися недостатньо ефективними у задачах, де шахрайські операції маскуються під легітимну активність або формують складні багаторівневі залежності.

Критичний аналіз сучасних досліджень свідчить, що класичні алгоритми мають обмежену здатність до узагальнення у випадках зміни концепції (concept drift) – зміни статистичних закономірностей у транзакційних потоках [3]. У результаті виникає необхідність регулярного оновлення моделей і використання більш адаптивних підходів до навчання.

Подальший розвиток пов'язаний із впровадженням підходів репрезентативного навчання (representation learning), що забезпечують автоматичне формування ознак із великих масивів транзакційних даних. Глибинні нейронні мережі дозволяють моделювати складні нелінійні залежності між характеристиками транзакцій і підвищують здатність систем виявляти слабко виражені ризикові патерни.

Значну увагу в сучасних роботах приділяють послідовним моделям, зокрема трансформерним архітектурам і темпоральним згортковим мережам, які враховують часову структуру фінансової активності клієнтів. Такі підходи демонструють покращення показників виявлення шахрайства порівняно з традиційними алгоритмами, особливо у задачах прогнозування ризикових дій

користувачів [2]. Разом із тим підвищення складності моделей призводить до зростання вимог до обчислювальних ресурсів і ускладнює їх інтеграцію у реальні фінансові системи. Це актуалізує проблему пошуку компромісу між точністю та продуктивністю.

Одним із найбільш перспективних напрямів є застосування графових нейронних мереж, які дозволяють враховувати мережеву природу фінансових взаємодій. У транзакційних графах вузли можуть відповідати рахункам, користувачам або пристроям, а ребра – фінансовим операціям. Дослідження показують, що графові моделі здатні виявляти організовані шахрайські схеми, які залишаються непомітними для моделей, що аналізують транзакції ізольовано [1]. Подальший розвиток пов’язаний із використанням динамічних графових архітектур, які враховують еволюцію фінансових мереж у часі та підвищують адаптивність систем до нових стратегій шахрайства [4].

У задачах фінансового шахрайства актуальною залишається проблема значної нерівномірності розподілу класів. У зв’язку з цим активно досліджуються підходи виявлення аномалій, що дозволяють моделювати нормальну поведінку транзакційних потоків і фіксувати відхилення від неї.

Крім того, перспективним напрямом є використання федеративного навчання (federated learning), яке дозволяє кільком фінансовим установам спільно навчати моделі без обміну конфіденційними даними. Такий підхід сприяє формуванню більш узагальнених моделей і підвищує рівень інформаційної безпеки [5].

Сучасні системи виявлення шахрайства будуються як потокові аналітичні платформи, інтегровані у фінансові сервіси. Їхня архітектура передбачає використання модулів генерації ознак у реальному часі, сервісів прогнозування та підсистем моніторингу якості моделей.

Одним із ключових викликів є забезпечення стабільності роботи моделей у production-середовищі в умовах зміни концепції. Це потребує впровадження адаптивних стратегій навчання, автоматичного донавчання моделей і

використання гібридних рішень, що поєднують механізми на основі правил з методами машинного навчання.

Сучасні методи машинного навчання суттєво розширюють можливості виявлення фінансового шахрайства, забезпечуючи аналіз великих потоків транзакційних даних і підвищуючи здатність систем адаптуватися до нових типів загроз. Найбільш перспективними напрямами розвитку є використання графових нейронних мереж, темпоральних моделей і підходів федеративного навчання. Водночас практична ефективність таких рішень визначається не лише якістю алгоритмів, а й їх здатністю функціонувати у реальних фінансових системах із жорсткими вимогами до швидкодії, інтерпретованості та стабільності метрик.

Список використаних джерел:

1. Dou Y., Liu Z., Sun L., Deng Y., Peng H., Yu P. S. Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters // *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. New York : ACM, 2020. P. 3158–3162. DOI: 10.1145/3340531.3411903.
2. Chen Y., Zhao C., Xu Y., Nie C., Zhang Y. Deep Learning in Financial Fraud Detection: Innovations, Challenges, and Applications // *Data Science and Management*. 2025. DOI: 10.1016/j.dsm.2025.08.002.
3. Gama J., Žliobaitė I., Bifet A., Pechenizkiy M., Bouchachia A. A Survey on Concept Drift Adaptation // *ACM Computing Surveys*. 2014. Vol. 46, No. 4. Article 44. DOI:10.1145/2523813.
4. Ren L., Hu R., Li D., Liu Y., Wu J., Zang Y., Hu W. Dynamic graph neural network-based fraud detectors against collaborative fraudsters // *Knowledge-Based Systems*. 2023. Vol. 278. Article 110888. DOI: 10.1016/j.knosys.2023.110888.
5. Yang Q., Liu Y., Chen T., Tong Y. Federated Machine Learning: Concept and Applications // *ACM Transactions on Intelligent Systems and Technology*. 2019. Vol. 10, No. 2. Article 12. P. 1–19. DOI: 10.1145/3298981.

УДК 004.89:004.7:005.934

ВИКОРИСТАННЯ ЛОКАЛЬНИХ ШІ-МОДЕЛЕЙ ДЛЯ ПІДВИЩЕННЯ КОНФІДЕНЦІЙНОСТІ В СИСТЕМАХ «РОЗУМНОГО ДОМУ»

Перехрест Д. О.
perekhrestdmytro07@gmail.com
Черкаський державний фаховий бізнес коледж
Ратайчук П. Є.
м.Черкаси, Україна

Сьогодні технології IoT фактично стали стандартом для сучасного житла, а системи «Розумного дому» зустрічаються все частіше. Величезна кількість датчиків та мікроконтролерів генерує постійний потік даних, який потребує розумної обробки. Зазвичай для цього використовують хмарні сервіси, але у них є три суттєві недоліки: залежність від стабільного інтернету, помітні затримки у реакції та питання конфіденційності.

Саме тому зараз стає популярним підхід Edge Computing, коли штучний інтелект працює не десь на віддаленому сервері, а безпосередньо в мережі будинку. Локальний запуск ШІ-моделей дозволяє зробити систему повністю автономною та швидкою, а головне – гарантує, що особисті дані не покинуть межі квартири.

Головною перевагою розгортання локальних великих мовних моделей (LLM) є абсолютна приватність. Жодна аудіокоманда чи телеметрія з датчиків не залишає меж фізичної мережі будинку. Крім того, локальні моделі забезпечують мінімальний час відгуку, що є критично важливим для систем реального часу, де затримка у ввімкненні освітлення чи спрацюванні сигналізації викликає дискомфорт або небезпеку.

Ефективність роботи інтелектуальних моделей безпосередньо залежить від технічних характеристик сервера. На відміну від хмарних сервісів, де обчислювальні ресурси практично необмежені, локальний запуск LLM потребує значного обсягу відеопам'яті (VRAM). Використання методів оптимізації, як-от квантування моделей до 4 або 8 біт, дозволяє успішно застосовувати споживчі графічні адаптери. Зокрема, відеокарта рівня NVIDIA RTX 3060 з 12 ГБ

відеопам'яті є оптимальним рішенням для розгортання моделей із кількістю параметрів 7–8 мільярдів. Це робить побудову інтелектуальної системи керування доступною, оскільки зникає потреба у дорогому професійному обладнанні.

Щоб адаптувати базову мовну модель до умов конкретної IoT-мережі, доцільно використовувати методи ефективного донавчання, зокрема технологію LoRA (Low-Rank Adaptation). Це дозволяє моделі «вивчити» унікальну топологію будинку, назви окремих пристроїв та логіку роботи мікроконтролерів (наприклад, на базі ESP32). У результаті ШІ перестає бути просто текстовим помічником: він починає генерувати точні структуровані команди у форматі JSON або передавати інструкції через протокол MQTT безпосередньо на виконавчі пристрої.

Можливості локального штучного інтелекту в межах розумного дому значно ширші за звичайне голосове керування. Система може виступати аналітичним центром, що здійснює предиктивний аналіз на основі даних із датчиків температури та присутності. Це дозволяє автоматично оптимізувати роботу кліматичної техніки, враховуючи історію енергоспоживання.

Окрім комфорту, локальна модель стає ключовим елементом безпеки. Вона здатна розпізнавати аномалії в поведінці системи – наприклад, зафіксувати нетиповий рух або вчасно попередити про залишені ввімкненими електроприлади. Важливо, що такий глибокий аналіз патернів життя мешканців відбувається повністю автономно, що виключає будь-яку можливість витоку конфіденційної інформації за межі домашнього сервера.

Окрему увагу в межах розробки системи моніторингу слід приділити архітектурі взаємодії між локальним інтелектуальним ядром та периферійними вузлами мережі. На відміну від стандартних комерційних рішень, пропонована система базується на використанні відкритих протоколів передачі даних. Використання мікроконтролерів сімейства ESP32 як кінцевих вузлів дозволяє реалізувати гнучку мережу датчиків, що обмінюються даними через протокол MQTT. Це забезпечує високу швидкість реакції та стійкість до втрати пакетів.

Центральний сервер, що базується на ОС Linux, виконує роль брокера та агрегатора даних, передаючи структуровану інформацію до локальної LLM для подальшого прийняття рішень.

Процес обробки інформації в такій системі можна розділити на три рівні: сенсорний рівень (збір первинних даних про температуру, вологість, освітлення та стан електромережі), рівень аналізу (обробка даних локальною моделлю штучного інтелекту) та виконавчий рівень (керування реле, освітленням чи побутовою технікою). Інтеграція голосового керування через локальні асистенти з відкритим кодом (наприклад, Jarvis) дозволяє реалізувати природний інтерфейс взаємодії без необхідності звернення до API зовнішніх сервісів. Це критично важливо для стабільної роботи системи в умовах можливих перебоїв з інтернет-зв'язком або при відключеннях електроенергії, коли локальна мережа продовжує функціонувати від джерел безперебійного живлення.

Додатковим фактором розширення функціональності є використання візуалізації даних моніторингу. Система передбачає побудову інтерактивних графіків споживання ресурсів, що дозволяє локальному ШІ надавати поради щодо енергоефективності. Наприклад, аналізуючи дані за певний період, модель може автоматично скоригувати графік роботи нагрівальних приладів, що призведе до значної економії коштів користувача без втрати рівня комфорту. Таким чином, система перетворюється з пасивного інструмента моніторингу на активного помічника.

Отже, впровадження локальних моделей штучного інтелекту в системи моніторингу та керування IoT-мережею «Розумний дім» є високоефективним рішенням, що дозволяє нівелювати ключові недоліки хмарних технологій. Розгортання таких систем забезпечує абсолютну конфіденційність користувацьких даних, мінімізує затримки при виконанні команд та гарантує автономну роботу навіть за відсутності підключення до мережі Інтернет. Практична розробка та тестування подібних IoT-мереж, зокрема інтеграція локальних голосових асистентів, підтверджує, що завдяки сучасним методам оптимізації та донавчання (LoRA, квантування) створення розумного та

безпечного житлового простору стає можливим на базі доступного комп'ютерного обладнання. Це відкриває нові перспективи для створення надійних та персоналізованих систем автоматизації.

Список використаних джерел:

1. Бакурова А. В., Терещенко О. В. Інтернет речей: технології та застосування : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 245 с.
2. Олійник В. В. Периферійні обчислення (Edge Computing) в системах розумного дому: переваги та виклики // Сучасні інформаційні технології та комп'ютерна інженерія. 2024. Вип. 15. С. 112–118.
3. Коваленко О. М. Архітектура розподілених систем управління на базі мікроконтролерів ESP32 та протоколу MQTT // Вісник комп'ютерних систем та мереж. 2023. № 4. С. 45–52.
4. LoRA: Low-Rank Adaptation of Large Language Models / E. J. Hu, Y. Shen, P. Wallis [та ін.] // arXiv preprint. 2021. URL: https://arxiv.org/abs/2106.09685 (дата звернення: 16.03.2026).
5. Офіційна документація платформи Home Assistant. URL: https://www.home-assistant.io/docs/ (дата звернення: 16.03.2026).

УДК 004.056.53:004.8

МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ МЕРЕЖЕВИХ ВТОРГНЕНЬ У СИСТЕМАХ КІБЕРБЕЗПЕКИ

Олійник М.С.
maksres01@gmail.com
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м. Черкаси, Україна

У сучасному середовищі інформаційних технологій динаміка розвитку кіберзагроз вимагає впровадження інтелектуальних засобів захисту. Традиційні системи виявлення вторгнень (IDS), що базуються на сигнатурному аналізі, демонструють обмежену ефективність проти атак нового типу, оскільки вони не

здатні розпізнавати аномалії, що відсутні у заздалегідь визначених базах даних. Саме тому застосування методів машинного навчання є пріоритетним напрямком, що дозволяє автоматизувати процес ідентифікації підозрілої активності на основі аналізу статистичних характеристик мережевих потоків [2].

Для побудови ефективної моделі розглядається архітектура системи, що базується на обробці великих масивів даних у режимі реального часу. Важливим етапом проектування є вибір репрезентативного набору даних для навчання. У ході дослідження проаналізовано характеристики сучасного датасету NSL-KDD, який є вдосконаленою версією KDD Cup 99 та містить записи нормальної активності та основних категорій атак, таких як відмова в обслуговуванні та несанкціоноване сканування [4]. Використання даного набору дозволяє уникнути надмірності даних, що позитивно впливає на якість навчання класифікаторів та швидкість обробки інформації модулями виявлення.

Особлива увага при реалізації системи приділяється етапу попередньої обробки та виділення найбільш інформативних ознак. Для точної ідентифікації вторгнень критично важливими визначено параметри тривалості сесії, типу протоколу та кількості помилкових пакетів [1]. Застосування методів нормалізації значень дозволяє збалансувати вплив різних ознак на кінцевий результат, що підвищує стабільність роботи алгоритмів у гетерогенних мережевих середовищах. Процес підготовки даних реалізується засобами мови програмування Python із використанням спеціалізованих бібліотек для обробки статистичної інформації.

У процесі аналізу алгоритмів машинного навчання встановлено, що використання ансамблевих методів, зокрема випадкового лісу, дозволяє досягти високої точності класифікації за рахунок побудови сукупності дерев рішень. Такий підхід суттєво знижує ризик перенавчання моделі та забезпечує надійне розпізнавання аномалій навіть у складних структурах трафіку [1]. Крім того, досліджено можливості методу опорних векторів, який демонструє високу ефективність у задачах розділення трафіку на легітимний та шкідливий за умови коректного вибору ядерної функції та параметрів регуляризації [3].

Експериментальна перевірка системи підтверджує, що обрані математичні підходи дозволяють виявляти вторгнення з точністю понад 97%. Програмна реалізація системи передбачає модульну структуру, що дозволяє інтегрувати її в існуючі комплекси моніторингу мережевої активності. Оцінка продуктивності моделі проводилася на основі метрик повноти та точності, що підтвердило здатність системи мінімізувати кількість хибнопозитивних спрацювань. Це створює умови для ефективного розгортання інтелектуальних модулів у корпоративних мережах з метою забезпечення проактивного захисту інформаційних ресурсів.

Таким чином, інтеграція методів машинного навчання в архітектуру систем захисту інформації дозволяє створити адаптивний та гнучкий механізм виявлення загроз. Запропонований підхід забезпечує стабільність функціонування мережі та автоматизує процес розпізнавання нових типів атак, що є критично важливим для сучасних систем кібербезпеки. Подальші дослідження у даному напрямку можуть бути спрямовані на вдосконалення моделей для аналізу зашифрованого трафіку та використання методів глибокого навчання для підвищення швидкодії системи в умовах високої інтенсивності передачі даних.

Список використаних джерел:

1. Бурячок В. Л., Толубко В. Б. Інформаційна та кібербезпека: соціотехнічний аспект. Київ: ДУТ, 2019. 432 с.
2. Гнатуш А. В. Методи та засоби машинного навчання в кібербезпеці: навч. посіб. Львів : Видавництво Львівської політехніки, 2022. 180 с.
3. Сучасні інтелектуальні системи захисту інформації в комп'ютерних мережах. URL: https://infosec.org.ua/intellectual-ids (дата звернення: 01.03.2026).
4. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2006. 738 p.

NSL-KDD Dataset for Network Intrusion Detection. University of New Brunswick. URL: https://www.unb.ca/cic/datasets/nsl.html (дата звернення: 08.03.2026).

УДК 004.8:004.93

ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ БПЛА

Монько С.Ю.
stasmonko6@gmail.com
Черкаський державний фаховий бізнес-коледж
Швиденко А.В
м. Черкаси, Україна

У сучасних інформаційних системах важливим напрямом розвитку є автоматизований аналіз відеоданих, отриманих із безпілотних літальних апаратів (БПЛА). Такі системи застосовуються передусім у військовій сфері, а також для моніторингу територій, аналізу дорожнього руху, контролю інфраструктури, екологічного спостереження та інших прикладних задач [1].

Особливістю відеопотоку з БПЛА є висока динамічність сцени, зміна ракурсу, вплив погодних умов, коливання освітлення та наявність шумів. Це ускладнює процес розпізнавання об'єктів і вимагає використання ефективних алгоритмів комп'ютерного зору, здатних працювати в реальному часі [1, 2].

Одним із найбільш результативних підходів є використання згорткових нейронних мереж, які забезпечують автоматичне вилучення ознак із зображення та їх класифікацію. Сучасні моделі дозволяють ефективно ідентифікувати об'єкти навіть за умов складного фону та часткового перекриття [2].

Серед алгоритмів детекції об'єктів особливе місце займає сімейство моделей YOLO (You Only Look Once), що орієнтоване на обробку відеопотоку в реальному часі. Модель YOLOv8 вирізняється оптимізованою архітектурою та високою швидкодією при збереженні достатнього рівня точності [3].

Сучасна версія моделі YOLOv8 характеризується покращеною архітектурою та оптимізованими алгоритмами обробки. Архітектура моделі включає три основні компоненти (рис. 1): backbone, neck та head. Backbone

відповідає за вилучення ознак із вхідного зображення шляхом послідовного застосування згорткових шарів. Neck забезпечує об’єднання багаторівневих ознак для підвищення якості детекції об’єктів різних розмірів. Head виконує остаточне визначення координат об’єктів та їх класифікацію [3].

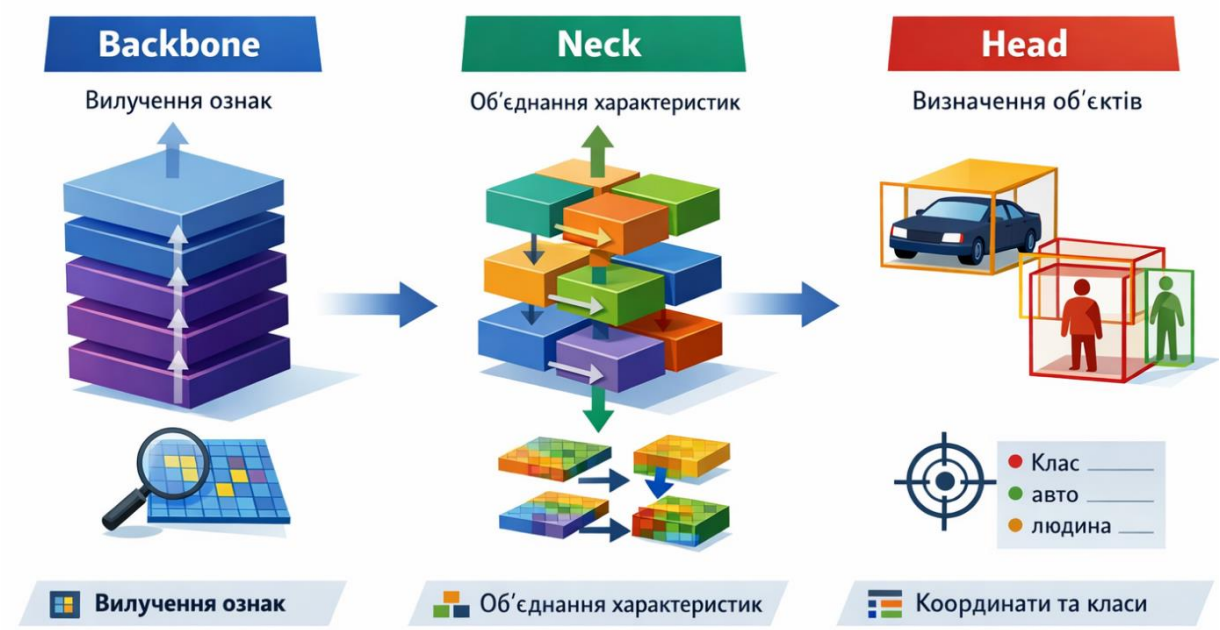


Рисунок 1 – Компоненти моделі YOLOv8.

Для вирішення проблеми обмежених обчислювальних ресурсів бортових систем доцільним є використання підходу з передачею відеопотоку на наземні або хмарні обчислювальні платформи. Це дозволяє застосовувати більш складні моделі без втрати продуктивності та зменшує вимоги до апаратного забезпечення БПЛА [3, 4].

Запропонований підхід передбачає багатоступеневу обробку відеоданих, що включає попередню обробку (фільтрація шуму, нормалізація), детекцію об’єктів, постобробку результатів та їх візуалізацію. Використання декількох спеціалізованих моделей, навчених на різних наборах даних, дозволяє підвищити точність розпізнавання в умовах змінного середовища [4].

Результати експериментальних досліджень показали, що система забезпечує обробку відеопотоку в реальному часі з частотою до 30–40 кадрів за секунду. При збільшенні кількості об’єктів у кадрі спостерігається зниження

швидкодії, однак система зберігає стабільність роботи та прийнятний рівень точності. Також встановлено залежність точності від висоти зйомки, роздільної здатності відео та умов освітлення [4].

Отримані результати підтверджують ефективність застосування нейронних мереж для аналізу відеопотоку з БПЛА. Запропонований підхід є універсальним інструментом комп'ютерного зору і може застосовуватись у різних сферах, зокрема у системах відеоспостереження, моніторингу та аналізу даних. Конкретна область застосування визначається умовами використання [5].

Перспективи подальших досліджень полягають у підвищенні точності моделей, оптимізації обчислювальних витрат та інтеграції систем комп'ютерного зору з іншими компонентами інтелектуальних інформаційних систем.

Таким чином, використання нейронних мереж для розпізнавання об'єктів у відеопотоці БПЛА є ефективним інструментом підвищення якості автоматизованого аналізу даних та розвитку сучасних інформаційних технологій.

Список використаних джерел:

1. Szeliski R. Computer Vision: Algorithms and Applications. Springer, 2022.
2. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016.
3. Jocher G. et al. YOLOv8 Documentation. Ultralytics, 2023.
4. Zhang Z. et al. Deep Learning for Object Detection: A Review // IEEE Access. 2021.
5. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. 2018.

УДК 004.8:004.932

ВІДНОВЛЕННЯ ПОШКОДЖЕНИХ ФОТОГРАФІЙ ЗА ДОПОМОГОЮ ГЕНЕРАТИВНИХ МОДЕЛЕЙ

Нечко Д. С.
dimanechkoq11@gmail.com
Черкаський державний фаховий бізнес-коледж
Ночевнов Д. П.
м. Черкаси, Україна

Відновлення пошкоджених фотографій (інпейтинг) є однією з важливих задач комп'ютерного зору. Воно полягає у заповненні відсутніх або пошкоджених ділянок зображення з використанням контекстної інформації [1]. З масовим використанням методів відновлення, зростає і попит на високу якість фото та відео в інтернеті. Значний вплив на розвиток генеративного синтезу зображень мали генеративно-змагальні мережі (GAN). Їх принцип роботи полягає в одночасному навчанні двох нейромереж: генератора і дискримінатора, які складаються з Convolutional Neural Network (CNN) Задача дискриминатора полягає у виявлені реальних та фальшивих даних, а генератора – генерувати такі фото щоб дискриминатор не зміг їх відрізнити від реальних. Водночас GAN-моделі мають обмеження у відновленні складних структур (текстур, просторових відношень) та підтриманні семантичної цілісності, особливо при роботі з невирівняними обличчями чи дрібними деталями, що призводить до появи артефактів і отримання не коректних результатів [1, 2].

Іншим перспективним напрямком є прискорення процесу генерації. Прикладом такого підходу є модель TurboFill. Її архітектура належить до класу diffusion-based models і пропонує вирішення ключової проблеми інпейнтингу – забезпечення одночасної структурної та семантичної узгодженості відновлених областей . Основною інновацією PixelHacker є механізм керування латентними категоріями, що отримав назву Latent Categories Guidance (LCG). Цей механізм базується на роздільному кодуванні ознак переднього плану (foreground) та фону (background) [1].

Архітектура PixelHacker будується на основі latent diffusion, де процес денойзингу модифіковано для інтеграції LCG-ембедингів. На відміну від традиційних підходів, де ознаки обробляються узагальнено, запропонована архітектура забезпечує розмежування відповідальності: foreground embeddings відповідають за цілісність об'єктів, тоді як background embeddings контролюють узгодженість оточення. Така декомпозиція дозволяє уникнути типових для інпейнтингу артефактів, пов'язаних зі змішуванням семантично різнорідних елементів під час відновлення втрачених областей [1].

Окрему увагу привертають моделі, орієнтовані на роботу на пристроях з обмеженими ресурсами. Прикладом є модель KXT-GAN (Kernel Expansion Transform GAN), представлена в роботі [3], яка пропонує архітектурне вирішення ключової проблеми впровадження систем відновлення зображень безпосередньо на смартфонах споживчого класу. Головний виклик полягає в обмежених обчислювальних ресурсах цих пристроїв, що унеможливлює пряме використання важких генеративних моделей, які зазвичай розгортаються на серверах.

Основною інновацією KXT-GAN є спеціалізована архітектура, орієнтована на ефективне виконання на нейронних процесорах (NPU) сучасних мобільних платформ. В її основі лежить новий тип згортки – Depth-dilated Fusion Convolution (DDF-Conv) [3]. Зменшення параметрів за допомогою DDF-Conv може обмежити здатність моделі розуміти довгострокові контекстуальні залежності, необхідні для якісного заповнення великих пошкоджених областей. Для компенсації використовується Kernel Expansion Transform Block.

Ще однією особливістю є використання крос-рівневих з'єднань, подібних до архітектури U-Net. Ці з'єднання допомагають ефективно передавати дрібнозернисті ознаки з енкодера до декодера, мінімізуючи втрату важливої просторової інформації. Це є ключовим для уникнення артефактів, таких як повторювані текстури чи кольорові плями, особливо при роботі з високою роздільною здатністю та великими областями дефектів [4].

Перш за все, відновлення фото активно використовуються для реставрації пошкоджених фотографій, зокрема для усунення подряпин, плям, заломів та інших дефектів, що виникли внаслідок фізичного старіння знімків або помилок під час їхнього зберігання [1, 2]. У цьому контексті генеративні моделі дозволяють не просто зафарбувати дефект, а й семантично узгоджено домальовувати втрачені фрагменти, спираючись на контекстну інформацію з непошкоджених ділянок [1].

Іншою важливою сферою є редагування цифрових зображень у професійних та аматорських фоторедакторах, де інпейнтинг використовується для видалення небажаних об'єктів (наприклад, випадкових перехожих, сміття чи технічних елементів) або для відновлення фону після вирізання певного об'єкта [2].

Сучасні архітектури, що орієнтовані на роботу на смартфонах споживчого класу, дозволяють виконувати такі операції безпосередньо на пристрої без передачі даних на сервер, що критично для збереження конфіденційності приватних знімків [3]. Крім того, технології відновлення знаходять застосування в кіноіндустрії та мультимедіа для ремастерингу старих відеоматеріалів, усунення артефактів стиснення та підвищення якості зображень у реальному часі [4]. У наукових дослідженнях методи інпейнтингу застосовуються, зокрема, для обробки супутникових знімків і видалення хмар під час аналізу рельєфу (рис.1), тоді як GAN-моделі можуть використовуватися для візуалізації можливих змін у міському середовищі (рис. 2) [5].

Розглянувши сучасні підходи до відновлення зображень демонструють чітку еволюцію від класичних генеративних мереж до більш спеціалізованих архітектур, кожна з яких вирішує специфічні задачі. Сучасні методи інпейнтингу розвиваються у напрямку покращення структурної та семантичної узгодженості, підвищення швидкодії через скорочення кроків дифузії [3], а також адаптації до роботи на пристроях з обмеженими ресурсами, зокрема смартфонах [4].

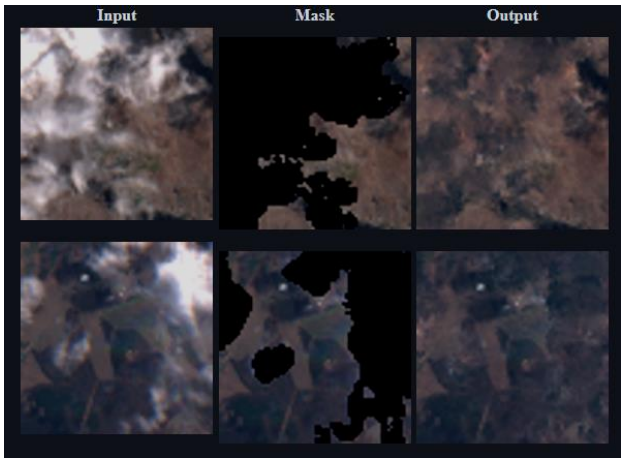


Рисунок 1 – Видалення хмар

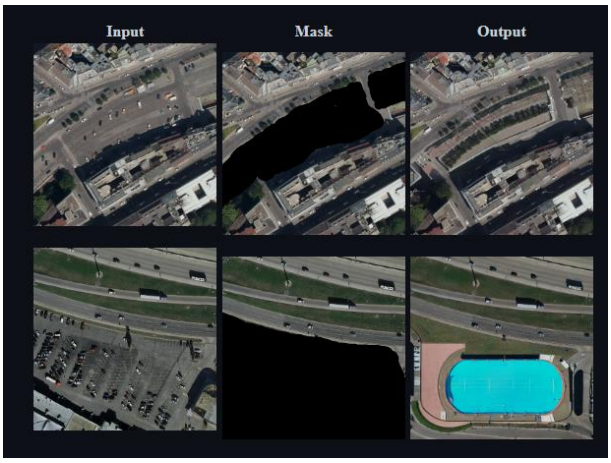


Рисунок 2 – Приклад зміни у місті

Незважаючи на відмінності в архітектурних рішеннях – від керування латентними категоріями до легких згорток та адаптерних систем – спільною рисою цих підходів є прагнення до балансу між якістю відновлення, обчислювальною ефективністю та збереженням контекстуальної цілісності зображень.

Список використаних джерел:

1. PixelHacker: image inpainting with structural and semantic consistency / Z. Xu та ін. // arXiv. 2025. URL: https://arxiv.org/abs/2504.20438 (дата звернення: 15.03.2026).
2. PixelHacker Project Page. HUST Vision Lab. URL: https://hustvl.github.io/PixelHacker/ (дата звернення: 15.03.2026).
3. Multi-Scale receptive field architecture for consumer-grade smartphone image inpainting/M. Tan та ін. // TechRxiv. URL: https://www.techrxiv.org/users/911592/articles/1285179 (дата звернення: 15.03.2026).
4. GitHub - htyjers/NTN-Diff: [NeurIPS 2025] One Stone with Two Birds: A Null-Text-Null Frequency-Aware Diffusion Models for Text-Guided Image Inpainting. GitHub. URL: https://github.com/htyjers/NTN-Diff?tab=readme-ov-file (дата звернення: 15.03.2026).

5. Diffusion Models for Earth Observation Use-cases: from cloud removal to urban change detection. ar5iv. URL: https://ar5iv.labs.arxiv.org/html/2311.06222 (дата звернення: 15.03.2026).

УДК 004.8:070:004.056

АВТОМАТИЗОВАНИЙ АНАЛІЗ ФЕЙКОВИХ НОВИН ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ

Воробйова В.Ю.
vorobyova.valentina2005@gmail.com
Черкаський державний фаховий бізнес коледж
Бреус Р.В.
м. Черкаси, Україна

Фейкові новини стали серйозною загрозою для сучасного інформаційного середовища, оскільки можуть маніпулювати громадською думкою, поширювати дезінформацію та створювати соціальну напругу. Ручні методи перевірки фактів вимагають значних ресурсів і часу, тому автоматизовані підходи, що базуються на штучному інтелекті, є перспективним рішенням для виявлення неправдивої інформації. Завдяки сучасним алгоритмам штучного інтелекту можна аналізувати великі обсяги даних, визначати ненадійні джерела та виявляти маніпуляції у текстах, зображеннях і відео.

Штучний інтелект використовує різні підходи для аналізу та виявлення фейкових новин. Один із ключових методів – обробка природної мови (Natural Language Processing, NLP), що дозволяє аналізувати структуру та зміст тексту. Алгоритми штучного інтелекту виявляють характерні ознаки неправдивих новин, такі як емоційно забарвлена лексика, упередженість чи сенсаційність. Завдяки глибокому навчанню нейромережі здатні розпізнавати мовні патерни та логічні невідповідності, що часто присутні у фейкових матеріалах.

Інший важливий аспект – перевірка достовірності джерел. Алгоритми аналізують репутацію джерела, наявність надійних посилань та кореляцію з іншими авторитетними медіа. Для цього використовуються великі бази даних, які містять інформацію про довіреність різних новинних порталів. Окрім

текстового аналізу, ШІ здатен розпізнавати змінені зображення та відео за допомогою алгоритмів комп’ютерного зору. Це особливо актуально у боротьбі з так званими "deepfake"-технологіями, які дозволяють створювати відео, що візуально ідентичні реальним, але містять неправдивий зміст [1].

Сучасні методи боротьби з дезінформацією ґрунтуються на кількох технологічних підходах. По-перше, застосовуються алгоритми машинного навчання (Machine Learning, ML) та глибокого навчання (Deep Learning, DL). Такі моделі навчаються на великих наборах даних, де аналізують мільйони статей, соціальних постів і коментарів для виявлення спільних рис фейкових новин. Нейромережі можуть розпізнавати структуру речень, тональність тексту та логічну послідовність викладених фактів.

По-друге, активно розвивається метод Explainable AI (XAI), який робить результати аналізу штучного інтелекту більш зрозумілими для користувачів. Це дає змогу не лише виявити фейк, а й пояснити, чому певна новина класифікується як неправдива.

Також використовуються технології блокчейну для перевірки автентичності новин. Блокчейн дозволяє відстежувати походження інформації, запобігаючи маніпуляціям та зміні вмісту публікацій після їх розповсюдження [2].

Попри значні досягнення у сфері автоматизованого аналізу фейкових новин, існують певні виклики. Одним із головних є постійна адаптація методів створення фейкових новин, що змушує ШІ-моделі постійно оновлюватися та вдосконалюватися. Додатковою складністю є неоднозначність деяких новин, що містять змішані факти, що ускладнює їх класифікацію. Крім того, надмірне використання автоматизованих алгоритмів може призводити до блокування правдивої, але суперечливої інформації, що може викликати питання щодо цензури та свободи слова.

Майбутні дослідження у сфері автоматизованого аналізу фейкових новин можуть зосередитися на кількох напрямках. По-перше, важливо створювати більш точні та адаптивні моделі ШІ, які здатні швидко реагувати на нові методи

дезінформації. По-друге, інтеграція технологій блокчейну для перевірки джерел може забезпечити більшу прозорість і незмінність інформації. По-третє, необхідно розробляти інструменти пояснюваного ШІ (Explainable AI), які дозволять зробити аналіз прозорішим для користувачів [3].

У перспективі можливе створення глобальних платформ перевірки фактів, що поєднуватимуть штучний інтелект із експертною оцінкою журналістів та аналітиків. Це дозволить підвищити ефективність боротьби з дезінформацією та зробити цифровий інформаційний простір безпечнішим для користувачів.

Штучний інтелект відіграє ключову роль у боротьбі з фейковими новинами, забезпечуючи ефективний аналіз інформації, перевірку джерел і виявлення маніпуляцій. Однак для досягнення максимальної ефективності необхідно поєднувати автоматизовані методи з людським контролем та вдосконалювати алгоритми аналізу. Використання сучасних технологій, таких як машинне навчання, глибокі нейромережі, Explainable AI та блокчейн, може значно підвищити ефективність боротьби з фейками та забезпечити достовірність інформації в медіа-просторі.

Список використаних джерел:

1. BBC News Україна. Новини та матеріали за тематикою інформаційних технологій та медіа. URL: BBC News Україна (дата звернення: 18.03.2026).
2. Найновіші розробки алгоритмів для автоматичного розпізнавання фейкових новин // MIND UA. URL: MIND UA (дата звернення: 18.03.2026).
3. Що таке обробка природної мови? // Unite.AI. URL: Unite.AI (дата звернення: 18.03.2026).

УДК 004.8:004.056

МЕТОДИ ВИЯВЛЕННЯ ШКІДЛИВОГО КОНТЕНТУ, СТВОРЕНОГО ШІ

Садчиков В.О.
repvladika.ua@gmail.com
Черкаський державний фаховий бізнес-коледж
Захарова М.В.
м.Черкаси, Україна

Стрімкий розвиток генеративного штучного інтелекту дав змогу створювати реалістичні тексти, зображення, аудіо та відео, які дедалі важче відрізнити від справжніх. Це створює нові ризики для інформаційної безпеки, довіри до цифрових комунікацій і суспільної стабільності. Шкідливий контент, створений або змінений за допомогою ШІ, може використовуватися для дезінформації, шахрайства, підробки особи, маніпулювання громадською думкою та інших форм цифрового зловживання. Особливо небезпечним є те, що сучасні генеративні системи здатні імітувати зовнішність, голос, стиль мовлення та навіть окремі поведінкові риси людини. Унаслідок цього зловмисники можуть створювати переконливі фейкові повідомлення, аудіозаписи та відео, які важко відрізнити від справжніх[1]. Це створює серйозні фінансові, репутаційні та соціальні наслідки, тому розроблення та впровадження методів виявлення такого контенту є важливим завданням сучасної інформаційної безпеки.

Шкідливий контент, створений ШІ, можна розглядати як цифровий контент, що був згенерований або суттєво модифікований алгоритмами машинного навчання та може використовуватися для введення в оману, маніпуляції або поширення небезпечної інформації. Для протидії таким ризикам важливе значення має цифрова прозорість контенту, тобто можливість встановити, як саме було створено або змінено цифровий об’єкт, хто або що брало участь у його формуванні, а також чи були в нього внесені зміни після створення. У цьому контексті застосовують цифрові водяні знаки, метадані, цифрові підписи та інші засоби фіксації походження контенту[2].

Методи виявлення синтетичного контенту доцільно поділяти на три основні групи: виявлення за даними про походження, автоматизоване виявлення

на основі самого контенту та виявлення за участю людини. Ці підходи не виключають один одного й зазвичай дають найкращий результат у поєднанні. Водночас жоден із них не є абсолютно надійним, оскільки синтетичний контент може бути частково зміненим, проходити постобробку або спеціально створюватися з урахуванням уникнення детекції.

Виявлення шкідливого контенту, створеного або модифікованого за допомогою штучного інтелекту, є важливим напрямом забезпечення інформаційної безпеки та цифрової довіри. Розвиток генеративних технологій уможливив створення реалістичних текстів, зображень, аудіо та відео, що поряд із корисними застосуваннями підвищує ризики дезінформації, цифрового шахрайства, маніпуляцій і поширення небезпечних матеріалів. Жоден окремий технічний метод не є повним розв'язанням цієї проблеми, тому ефективність виявлення залежить від контексту використання, якості реалізації та належного контролю[3].

Першу групу становлять методи встановлення походження контенту. Вони базуються на записі та подальшій перевірці відомостей про джерело й історію цифрового об'єкта. До таких засобів належать цифрові водяні знаки, метадані та цифрові підписи. Цифрове водяне маркування передбачає вбудовування службової інформації безпосередньо у контент - зображення, відео, аудіо або текст. У водяному знаку можуть міститися відомості про походження, час створення чи інші характеристики матеріалу. Метадані, своєю чергою, дають змогу фіксувати автора, параметри створення та редагування, а цифрові підписи - перевіряти цілісність даних. Усе це допомагає підвищувати автентичність і достовірність цифрового контенту.

Водночас такі підходи мають обмеження. Метадані часто видаляються під час пересилання файлів або публікації на платформах, а водяні знаки можуть бути пошкоджені, видалені або підроблені. Крім того, наявність даних про походження ще не гарантує істинності змісту, а лише дає додатковий сигнал про джерело матеріалу.

Другу групу становлять автоматизовані методи контентного аналізу. Вони намагаються визначити, чи є контент синтетичним або модифікованим, аналізуючи сліди, що залишаються після генерації чи обробки. Для зображень і відео це можуть бути візуальні аномалії, неузгодженість освітлення, тіней і віддзеркалень, неприродні текстури, а також статистичні та форензічні сигнали, зокрема просторово-частотні характеристики. Для відео також аналізують рухи обличчя, моргання та інші поведінкові невідповідності.

Таблиця 1 – Інструменти та технології виявлення шкідливого контенту, створеного ШІ.

№	Назва інструмента	Домен	Модальність	Опис
1	ActiveFence	Виявлення	Текст, зображення, відео, аудіо	Інструмент виявлення AI-контенту для модерації та виявлення зловживань
2	AISEO	Виявлення	Текст	Визначає текст, написаний людиною, і текст, згенерований ШІ
3	Attestiv	Виявлення	Зображення, відео, документи	Валідація медіа та виявлення шахрайства
4	AudioSeal	Водяні знаки	Аудіо	Заснований на машинному навчанні інструмент для водяного маркування згенерованого мовлення, розроблений для ефективності
5	Azure AI Content Safety	Модерація контенту	Текст, зображення	Виявляє шкідливий користувацький і згенерований ШІ контент у застосунках і сервісах
6	Content Authenticity Initiative Signing Toolkit	Метадані	Зображення, відео, аудіо, документи	Інструмент для забезпечення автентичності контенту та відстеження його походження

Для аудіо виявлення ґрунтується на пошуку неприродних інтонацій, дефектів спектра, часових і частотних невідповідностей. Для тексту використовують інші характеристики: передбачуваність мовних конструкцій, стилістичну одноманітність, частоту повторів, а також показники як складність та нерівномірність тексту. Такі методи дають змогу знаходити ознаки штучної генерації безпосередньо у самому змісті матеріалу.

Третю групу формують підходи із залученням людини. У таких системах модератори, аналітики, фактчекери або інші фахівці перевіряють сумнівні результати та приймають рішення з урахуванням контексту. Це особливо важливо тоді, коли контент є частково синтетичним, коли потрібно оцінити спосіб його поширення, намір автора або наслідки можливої помилки. Поєднання автоматизованого аналізу з людською перевіркою зазвичай підвищує надійність оцінювання, хоча потребує більше часу та ресурсів[1].

Окреме місце в протидії шкідливому ШІ-контенту посідають превентивні та модераційні механізми. До них належать фільтрація навчальних даних, вхідних запитів і вихідного контенту, використання класифікаторів небезпечного вмісту, блок-листів ключових слів, а також хешування вже виявлених шкідливих матеріалів для їх повторного розпізнавання на різних платформах. Такі методи допомагають не лише виявляти небезпечний вміст, а й обмежувати його подальше поширення. Важливим при цьому є правильне налаштування порогів спрацювання, оскільки баланс між хибнопозитивними та хибнонегативними результатами безпосередньо впливає на ефективність системи.

Ще одним важливим аспектом є тестування та оцінювання систем виявлення. Для цього потрібно враховувати не лише формальні метрики точності, а й контекст використання, відтворюваність результатів, характер можливих помилок і стійкість до атак. Для систем із залученням людини важливо також оцінювати час виконання перевірки та складність роботи для користувача [4].

Таким чином, методи виявлення шкідливого контенту, створеного ШІ, охоплюють відстеження походження контенту, автоматизований аналіз його ознак та перевірку за участю людини. Найперспективнішим є не окремий метод, а їх поєднання: метадані й водяні знаки дають сигнал про походження, автоматизовані моделі масштабують аналіз, а людина допомагає врахувати контекст і зменшити ризик помилок. Водночас, жоден із цих підходів не є універсальним, тому подальший розвиток теми пов’язаний із удосконаленням

детекторів для різних мов і модальностей, виявленням частково синтетичного контенту, підвищенням стійкості до атак та розбудовою стандартів цифрової прозорості й міжплатформної взаємодії.

Список використаних джерел

1. «Шкідливий інтелект: Сценарії злочинного використання ШІ». URL:
2. https://my-itspecialist.com/adversarial-intelligence-malicious-scenarios-of-ai-usage (дата звернення: 18.03.2026).
3. «Reducing Risks Posed by Synthetic Content An Overview of Technical Approaches to Digital Content Transparency». URL: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-4.pdf (дата звернення: 18.03.2026).
4. «Виявлення контенту, створеного за допомогою ШІ». URL: https://gijn.org/ua/resurs-ua/viavlenna-kontentu-stvorenogo-za-dopomogou-si-posibnik-dla-zurnalistiv/ (дата звернення: 18.03.2026).
5. «Маркування ШІ-згенерованого контенту: як уряди та компанії посилюють прозорість використання штучного інтелекту в медіа та соцмережах». URL: https://dslua.org/publications/markuvannia-shi-zghenerovanoho-kontentu-iak-uriady-ta-kompanii-posyliuiut-prozorist-vykorystannia-shtuchnoho-intelektu-v-media-ta-sotsmerezhakh/ (дата звернення: 18.03.2026).

СЕКЦІЯ 2

ІНЖЕНЕРНІ ПІДХОДИ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.42:004.438

СУЧАСНІ ПІДХОДИ ДО КРОСПЛАТФОРМНОЇ РОЗРОБКИ

Авраменко С.О.
avramenkosiroza955@gmail.com
Черкаський державний фаховий бізнес коледж
Фальченко Н. Г.
м. Черкаси, Україна

У сучасних умовах стрімкого розвитку інформаційних технологій мобільні та комп'ютерні пристрої стали невід'ємною частиною повсякденного життя людини. Зростання кількості користувачів різних платформ, таких як Android, iOS та персональні комп'ютери, зумовлює необхідність створення програмного забезпечення, яке здатне ефективно функціонувати на різних операційних системах. У зв'язку з цим особливої актуальності набуває кросплатформний підхід до розробки додатків [2].

Кросплатформна розробка передбачає створення програмного продукту на основі єдиної кодової бази з можливістю його використання на різних платформах без значних змін. Такий підхід дозволяє суттєво скоротити витрати часу та ресурсів на розробку, тестування і підтримку програмного забезпечення, а також забезпечити швидший вихід продукту на ринок[5].

Сучасні інструменти та середовища розробки, зокрема ігрові рушії та фреймворки, надають широкі можливості для створення продуктивних і функціональних додатків із високою якістю користувацького досвіду[3; 4].

Особливого значення кросплатформність набуває у сфері розробки відеоігор, де важливими є не лише технічні характеристики, а й зручність керування, оптимізація та адаптація під різні пристрої[1]. Використання єдиної архітектури дозволяє розробникам створювати універсальні ігрові продукти, які можуть бути однаково ефективними як на мобільних пристроях, так і на персональних комп'ютерах[3].

Метою кваліфікаційної роботи є дослідження особливостей розробки кросплатформних додатків та створення власного ігрового продукту, що поєднує в собі сучасні підходи до програмування, дизайну та інтерактивності.

У рамках даної роботи розробляється кросплатформна піксельна 2D гра у жанрі платформера з елементами паркуру та бойової системи. Гра призначена для використання як на мобільних пристроях, так і на персональних комп'ютерах, що забезпечує її доступність для широкого кола користувачів.

Основою ігрового процесу є поєднання динамічного пересування (паркур) та бойових механік. Головною особливістю гри виступає унікальна система взаємодії персонажа з навколишнім середовищем за допомогою спеціальних ціпків, які за сюжетом є частиною його тіла. Вони виконують не лише функцію зброї, а й слугують інструментом для подолання перешкод, що безпосередньо впливає на ігрову механіку та стиль проходження[1]. Блок-схему алгоритму проходження рівня зображено на рис.1.

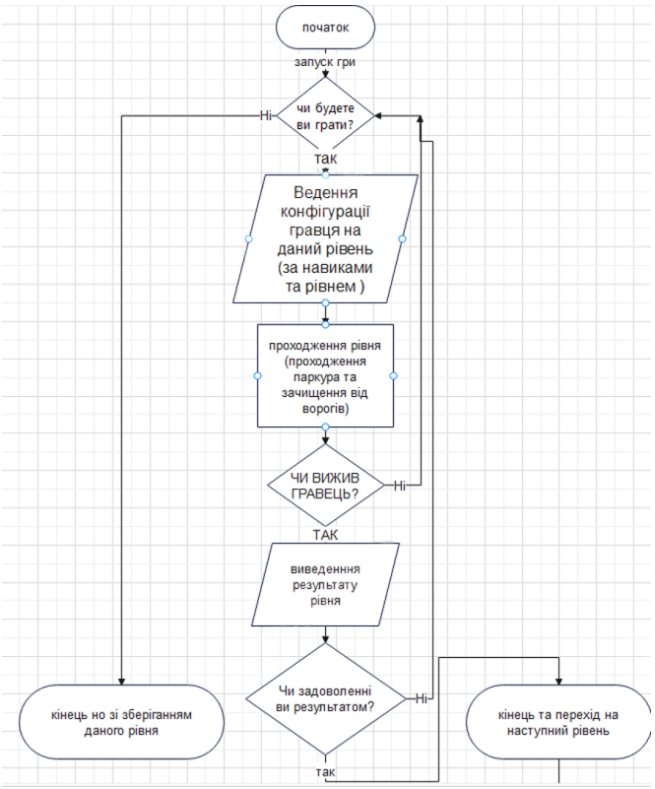


Рисунок 1 – Блок-схема алгоритму проходження рівня

У процесі гри користувач досліджує різні локації, які змінюються залежно від прогресу, отримує нові здібності, а також взаємодіє з різними персонажами та істотами, серед яких є як дружні, так і ворожі.

Важливим елементом гри є система розвитку персонажа та модифікації основної зброї. Ціпки можуть удосконалюватися, що дозволяє змінювати характеристики та стиль ведення бою. Це забезпечує варіативність ігрового процесу, підвищує інтерес користувача та сприяє формуванню індивідуального підходу до проходження гри[1].

У процесі виконання кваліфікаційної роботи передбачається вирішення таких завдань: аналіз сучасних підходів до кросплатформної розробки; вибір технологій та інструментів; проєктування архітектури програмного продукту; реалізація базової ігрової логіки, зокрема механік пересування (паркур) та бойової системи; створення кількох ігрових локацій (карт) для демонстрації основного функціоналу; розробка базового користувацького інтерфейсу; проведення тестування та первинної оптимізації додатка для різних платформ.

Об’єктом дослідження є процес розробки кросплатформних програмних продуктів, а предметом – методи та засоби створення 2D ігор із використанням сучасних технологій. Практичне значення роботи полягає у створенні ігрового прототипу, який може слугувати основою для подальшого розширення, вдосконалення та комерційної реалізації.

Таким чином, розроблений у межах кваліфікаційної роботи програмний продукт не є завершеним, а виступає початковим етапом більш масштабного проєкту. У подальшому передбачається розвиток гри шляхом додавання нових рівнів, механік, сюжетних елементів, систем розвитку персонажа та розширення функціональних можливостей, що дозволить перетворити її на повноцінний ігровий продукт.

Список використаних джерел:

1. Шелл Дж. Мистецтво геймдизайну : книга лінз. Київ : ArtHuss, 2021. 600 с.
2. Nystrom R. Game Programming Patterns. USA : Genever Benning, 2014. 354 p.
3. Unity Documentation. URL: https://docs.unity3d.com (дата звернення: 19.03.2026).
4. Godot Engine Documentation. URL: https://docs.godotengine.org (дата звернення: 19.03.2026).
5. React Native Documentation. URL: https://reactnative.dev (дата звернення: 19.03.2026).

УДК 004.8:004.415.2

ВИКОРИСТАННЯ AI-АСИСТЕНТІВ У ПРОЦЕСІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ: ПЕРЕВАГИ ТА РИЗИКИ

Мартиненко М. С.
martynenkoms2005@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р.В.
м. Черкаси, Україна

Стрімкий розвиток генеративного штучного інтелекту суттєво трансформує сучасний процес розробки програмного забезпечення. Якщо раніше автоматизація охоплювала переважно компіляцію, тестування та розгортання програмних продуктів, то сьогодні AI-асистенти беруть участь безпосередньо у створенні програмного коду, поясненні логіки алгоритмів, генерації тестів, документуванні, пошуку помилок і рефакторингу. До таких інструментів належать системи на основі великих мовних моделей, які інтегруються у середовища програмування та виконують роль інтелектуального помічника розробника.

Актуальність теми зумовлена тим, що використання AI-асистентів у програмній інженерії вже стало масовою практикою: за даними GitHub, понад 97% опитаних учасників команд розробки вказали, що хоча б раз

використовували AI coding tools у роботі, а за даними Stack Overflow, 76% респондентів уже використовують або планують використовувати AI-інструменти у процесі розробки [1; 6].

Однією з головних причин швидкого поширення AI-асистентів є їхній позитивний вплив на продуктивність праці програмістів. Такі системи дозволяють швидше створювати шаблонний код, пропонують готові фрагменти функцій, допомагають адаптуватися до нових мов програмування та спрощують розуміння великих кодових баз. У контрольованому експерименті Microsoft Research встановлено, що група розробників, яка використовувала GitHub Copilot, виконала поставлене завдання в середньому на 55,8% швидше порівняно з контрольною групою. Додатково GitHub зазначає, що AI-інструменти допомагають розробникам вивільняти час для більш складних видів діяльності, зокрема проєктування систем, співпраці в команді та глибшого аналізу вимог [1; 2].

Практична цінність AI-асистентів проявляється на різних етапах життєвого циклу програмного забезпечення. На етапі проєктування вони можуть узагальнювати функціональні вимоги, пропонувати структуру модулів і допомагати формувати архітектурні рішення. На етапі програмування AI-асистенти застосовуються для автодоповнення коду, генерації окремих класів, методів і SQL-запитів. У процесі тестування вони здатні формувати модульні тести та підказувати граничні сценарії перевірки. GitHub у своєму опитуванні також вказує, що понад 98% респондентів повідомили про експерименти їхніх організацій із використанням AI для генерації тестових випадків. Це свідчить про поступову інтеграцію AI-асистентів не лише в написання коду, а й у забезпечення якості програмних продуктів [1].

На рис.1 показано основні етапи життєвого циклу програмного забезпечення, на яких AI-асистенти можуть застосовуватися для автоматизації та інтелектуальної підтримки роботи розробника.

Рисунок 1 – Використання AI-асистентів на етапах життєвого циклу програмного забезпечення

Водночас ефективність використання таких інструментів не означає повної безпечності або безумовної якості результату. Одним із ключових ризиків є генерація синтаксично правильного, але логічно некоректного коду. Великі мовні моделі не “розуміють” програму у класичному інженерному сенсі, а статистично прогнозують найімовірніші послідовності токенів. Через це можливі помилки в алгоритмах, неправильне використання бібліотек, порушення бізнес-логіки, а також так звані галюцинації, коли система впевнено пропонує неіснуючі методи, API або параметри. Небезпека підсилюється тим, що згенерований код часто виглядає переконливо та може бути сприйнятий розробником як коректний без належної перевірки. Саме тому зростає значення експертної валідації результатів, статичного аналізу та обов’язкового код-рев’ю [5].

Окрему групу становлять ризики інформаційної безпеки. Якщо розробник передає AI-асистенту фрагменти закритого коду, конфігураційні файли, ключі доступу, логіни або внутрішню документацію, це може призвести до витоку чутливої інформації. OWASP серед критичних ризиків для LLM-застосунків виокремлює prompt injection, insecure output handling, supply chain vulnerabilities та sensitive information disclosure. У контексті розробки програмного

забезпечення це означає, що AI-асистент може стати додатковою поверхнею атаки: зловмисно сформований ввід здатний вплинути на відповіді моделі, а небезпечний або неперевірений вихідний код може бути вбудований у систему без достатнього контролю. Крім того, використання сторонніх моделей або плагінів створює ризики, пов’язані з ланцюгом постачання програмного забезпечення, коли небезпека виникає не лише в самому коді, а й у зовнішніх сервісах, від яких залежить команда розробки [4].

Проблемним залишається і питання довіри до AI-асистентів. Хоча значна частина розробників позитивно оцінює потенціал таких засобів, реальні результати їх упровадження є неоднозначними. Згідно зі звітом DORA 2024, більше 75% респондентів використовують AI щонайменше для одного щоденного професійного завдання, а понад третина повідомила про помітне зростання продуктивності. Водночас зростання рівня використання AI асоціювалося не лише з покращенням якості документації, коду та швидкості рев’ю, але й із погіршенням окремих показників software delivery: дослідники зафіксували оцінюване зниження пропускної здатності доставки на 1,5% та стабільності доставки на 7,2%. Це свідчить, що AI-асистенти не є універсальним рішенням і не можуть замінити зрілих інженерних практик, таких як автоматизоване тестування, невеликі батчі змін, контроль версій, безпечний CI/CD та архітектурний нагляд [3].

Для узагальнення основних переваг і ризиків використання AI-асистентів у процесі розробки програмного забезпечення доцільно подати їх у вигляді таблиці.

Дані, наведені в табл.1, свідчать, що AI-асистенти поєднують значний потенціал для підвищення продуктивності з низкою технічних і безпекових ризиків, що вимагає контрольованого використання таких інструментів.

Таблиця 1 – Основні переваги та ризики використання AI-асистентів у процесі розробки програмного забезпечення

№	Переваги	Ризики
1	Прискорення написання типового коду	Генерація некоректного коду
2	Автоматизація створення тестів	Поява прихованих вразливостей
3	Спрощення роботи з новими технологіями	Витік конфіденційних даних
4	Допомога в документуванні	Надмірна залежність від підказок
5	Підтримка рефакторингу	Зниження рівня критичного аналізу коду

У зв’язку з цим ефективне впровадження AI-асистентів має базуватися на принципі «людина в контурі прийняття рішень». AI повинен розглядатися не як автономний розробник, а як інструмент інтелектуальної підтримки, результати якого підлягають перевірці. NIST у профілі AI RMF для генеративного ШІ наголошує на потребі системного управління ризиками протягом усього життєвого циклу використання таких технологій. Для команд розробки це означає необхідність формування політик безпечного використання AI, заборони на передавання чутливих даних у зовнішні моделі, журналювання взаємодії з асистентами, обов’язкової ревізії згенерованого коду та навчання персоналу правилам безпечної роботи з генеративним ШІ [5].

Отже, AI-асистенти вже стали важливим елементом сучасної програмної інженерії та мають значний потенціал для підвищення продуктивності, якості документації, швидкості тестування та підтримки розробників у роботі з новими технологіями. Водночас їх використання супроводжується технічними, безпековими, організаційними та правовими ризиками. Тому найбільш раціональним підходом є не безумовне захоплення можливостями AI, а його контрольоване впровадження в межах інженерних стандартів, де автоматизовані рекомендації поєднуються з професійною відповідальністю фахівця. Подальші дослідження варто спрямувати на розробку методів оцінювання якості AI-

згенерованого коду, моделей довіри до AI-асистентів та практик їх безпечної інтеграції у життєвий цикл програмного забезпечення.

Список використаних джерел:

1. Survey: The AI wave continues to grow on software development teams // GitHub Blog. 2024. URL: https://github.blog/news-insights/research/survey-ai-wave-grows/ (дата звернення: 14.03.2026).
2. Peng S., Kalliamvakou E., Cihon P., Demirer M. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot // Microsoft Research. 2023. URL: https://www.microsoft.com/en-us/research/publication/the-impact-of-ai-on-developer-productivity-evidence-from-github-copilot/ (дата звернення: 15.03.2026).
3. Accelerate State of DevOps Report 2024 // DORA. 2024. URL: https://cloud.google.com/blog/products/devops-sre/announcing-the-2024-dora-report (дата звернення: 16.03.2026).
4. OWASP Top 10 for Large Language Model Applications // OWASP Foundation. 2025. URL: https://owasp.org/www-project-top-10-for-large-language-model-applications/ (дата звернення: 17.03.2026).
5. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1. 2024. URL: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf (дата звернення: 18.03.2026).
6. AI. 2024 Stack Overflow Developer Survey // Stack Overflow. 2024. URL: https://survey.stackoverflow.co/2024/ai (дата звернення: 18.03.2026).

УДК 004.4

ПІДСИСТЕМА ПУБЛІКУВАННЯ ОКРЕМИХ ТЕСТІВ НА DMOJ ЗА ЗАПИТАМИ КОРИСТУВАЧІВ

Шафієв Д. Ю.
shafiiev.daniil1122@vu.cdu.edu.ua
Черкаський національний університет
імені Богдана Хмельницького
Порубльов І. М.
м. Черкаси, Україна

Системи автоматичної перевірки задач з програмування стали важливою складовою підготовки майбутніх фахівців з інженерії програмного забезпечення, оскільки забезпечують автоматизовану перевірку розв'язків, об'єктивність оцінювання та зручну взаємодію між користувачем і платформою. Однією з відкритих систем такого типу є DMOJ – Modern Online Judge, що підтримує багато мов програмування, може бути розгорнута локально та допускає розширення функціональності за рахунок додаткових модулів [1].

Актуальність дослідження зумовлена тим, що більшість сучасних онлайн-суддів повністю або частково приховують тестові дані. Такий підхід є виправданим у змагальному середовищі, однак у навчальному процесі він ускладнює аналіз помилок і знижує ефективність налагодження програм. Користувач зазвичай бачить лише узагальнений результат перевірки, наприклад Wrong Answer або Time Limit Exceeded, але не має змоги встановити, які саме вхідні дані спричинили помилку. Тому розробка механізму контрольованого надання окремих тестів за запитами користувачів є актуальним практичним завданням.

Метою роботи є розробка підсистеми публікування окремих тестів у системі DMOJ за запитами користувачів шляхом створення програмного модуля з механізмами керування доступом. Запропоноване рішення має поєднати дві вимоги: надати користувачеві більше даних для аналізу власного розв'язку та водночас не допустити довільного розкриття повного набору перевірочних матеріалів [1].

На першому етапі дослідження було проаналізовано сучасні системи автоматичної перевірки задач з програмування та підходи до керування тестовими даними. Доцільно розмежовувати централізовані онлайн-платформи, зокрема LeetCode, HackerRank та EOlymp [4–6], і open-source judge-системи, серед яких DMOJ, DOMjudge, Judge0 та PC² [1–3; 7]. Централізовані платформи надають готовий сервіс, але не дають змоги змінювати внутрішню логіку перевірки або впроваджувати власні підсистеми. Натомість open-source рішення можуть бути розгорнуті у власній інфраструктурі, підтримують адаптацію архітектури та є придатними для створення спеціалізованих розширень [1–3; 7].

Проведений аналіз показав, що жодна з розглянутих систем у типовій конфігурації не забезпечує гнучкого механізму вибіркового публікування окремих тестів за запитами користувачів. Водночас саме DMOJ має архітектурні передумови для реалізації такої функціональності: відкритий код, наявність ролей користувачів, інтеграцію із серверною логікою перевірки та роботу з файловою системою тестових даних [1]. Саме тому цю платформу обрано як базове середовище для проєктування та подальшої реалізації підсистеми.

На другому етапі було сформовано вимоги до підсистеми публікування окремих тестів. Моделювання бізнес-процесу показало, що новий функціонал не повинен змінювати стандартний механізм оцінювання посилки, а має розширювати сценарій роботи зі сторінкою результатів. Після завершення перевірки та виявлення неуспішних тестових випадків користувач повинен мати змогу ініціювати запит лише для конкретного тесту, для якого публікування дозволено встановленими правилами.

На основі користувацьких історій визначено ключові функціональні вимоги до підсистеми. Вона повинна відображати доступність запиту лише для тих тестів, які одночасно є неуспішними в межах конкретної посилки та позначені як відкриті до публікування. Після натискання відповідної кнопки система має автоматично визначити задачу, знайти потрібний ZIP-архів тестових даних, отримати вхідні дані та еталонний вихід для вибраного тестового випадку, додати фактичний результат користувача й сформувати текстовий файл для

завантаження. У результаті користувач отримує звіт, придатний для подальшого аналізу, без додаткових ручних дій.

Окрему увагу приділено нефункціональним вимогам. Підсистема повинна бути безпечною, тобто не допускати довільного доступу до файлів поза встановленою структурою архівів і не дозволяти публікування даних для тестів, які не відповідають визначеним умовам. Важливими також є продуктивність, надійність, сумісність із наявною архітектурою DMOJ, зручність використання та супроводжуваність [1]. Це означає, що формування файлу має відбуватися без помітної затримки для користувача, а внутрішня структура модуля повинна допускати подальше розширення правил доступу та формату звіту.

Для підтримки запропонованого функціоналу обґрунтовано розширення моделі даних. Підсистема спирається на вже наявні у DMOJ сутності користувача, задачі, посилки, тестового випадку та результату виконання окремого тесту [1]. Разом із цим до логічної схеми доцільно додати окрему таблицю правила публікування тесту, яка зберігатиме ідентифікатор тестового випадку, ознаку дозволу на публікування, момент останнього оновлення та користувача, який виконав зміну. Такий підхід дає змогу інтегрувати нову функціональність без дублювання наявних даних і без порушення стандартної логіки роботи платформи.

Отже, у результаті проведеного дослідження обґрунтовано доцільність розробки підсистеми публікування окремих тестів на DMOJ за запитами користувачів, визначено її місце в архітектурі платформи та сформовано основні вимоги до реалізації. Запропоноване рішення має практичне значення для освітнього процесу, оскільки поєднує контрольований доступ до тестових даних із збереженням принципів чесного оцінювання [1]. Упровадження такої підсистеми сприятиме глибшому аналізу помилок, підвищенню якості студентських розв’язків і розширенню функціональних можливостей сучасних систем автоматичної перевірки програм.

Список використаних джерел:

1. DMOJ: Modern Online Judge. URL: https://dmoj.ca/ (дата звернення: 20.03.2026).
2. DOMjudge. Programming Contest Jury System. URL: https://www.domjudge.org/ (дата звернення: 20.03.2026).
3. Judge0. Open-source online code execution system. URL: https://judge0.com/ (дата звернення: 20.03.2026).
4. LeetCode. The world’s leading online programming learning platform. URL: https://leetcode.com/ (дата звернення: 20.03.2026).
5. HackerRank. Online coding tests and technical interviews. URL: https://www.hackerrank.com/ (дата звернення: 20.03.2026).
6. EOlymp. Website dedicated to competitive programming, algorithms and problem solving. URL: https://eolymp.com/ (дата звернення: 20.03.2026).
7. PC² Contest Control System. URL: https://pc2ccs.github.io/ (дата звернення: 20.03.2026).

УКД 004.415.2

АРХІТЕКТУРНІ ПІДХОДИ ДО ПРОЄКТУВАННЯ ЧАТ-БОТІВ У СУЧАСНИХ ПРОГРАМНИХ СИСТЕМАХ

Антоненко А.Я
annantonenko4567@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В.Ю.
м. Черкаси, Україна

Активний розвиток технологій штучного інтелекту, обробки природної мови та хмарних обчислень сприяв впровадженню чат-ботів у сучасні програмні системи. Сьогодні такі рішення використовуються для автоматизації взаємодії з користувачами, надання довідкової інформації, підтримки клієнтів та виконання типових операцій у веб-сервісах, мобільних застосунках і корпоративних платформах. У зв’язку з цим особливого значення набуває архітектурне проєктування чат-ботів, оскільки саме структура програмної системи визначає її

масштабованість, здатність до інтеграції з іншими сервісами та можливість подальшого розвитку функціоналу. Метою дослідження є аналіз сучасних архітектурних підходів до розробки чат-ботів і визначення ключових принципів побудови їх програмної структури [1].

Чат-боти можна розглядати як багаторівневі програмні системи, що забезпечують автоматизований діалог між користувачем і інформаційною платформою. У більшості сучасних рішень архітектура формується на основі модульного підходу, де кожен компонент виконує окрему функцію у процесі обробки запиту. Типова структура системи включає інтерфейс взаємодії з користувачем, модуль обробки природної мови, механізм керування діалогом, підсистему доступу до даних та модулі інтеграції із зовнішніми сервісами. Такий підхід дозволяє відокремити логіку обробки запитів від інтерфейсної частини та спрощує масштабування системи при зростанні навантаження [1; 2].

Інтерфейс взаємодії забезпечує приймання повідомлень користувача та передавання результату назад у вибраний канал комунікації. У сучасних чат-ботах цей рівень часто підтримує кілька платформ одночасно: веб-чат, мобільний застосунок або популярні месенджери. Архітектурно інтерфейс виконує роль адаптера, що перетворює повідомлення у стандартизований формат для подальшої обробки системою. Це дозволяє інтегрувати додаткові канали взаємодії, не змінюючи внутрішню архітектуру чат-бота.[2].

Важливою складовою архітектури є модуль обробки природної мови. Він аналізує текст повідомлення, визначає намір користувача та виділяє ключові елементи запиту. Отримані результати передаються до модуля керування діалогом, який визначає подальшу логіку взаємодії. У межах архітектури цей компонент зазвичай реалізується як окремий сервіс, що дозволяє змінювати або вдосконалювати мовні моделі без впливу на інші частини системи. Подібна ізоляція функцій є важливою умовою гнучкості програмної архітектури.

Керування діалогом забезпечує визначення сценарію відповіді та координацію взаємодії між різними модулями системи. Саме тут формується рішення про те, чи потрібно звернутися до бази знань, виконати запит до

зовнішнього сервісу або згенерувати відповідь на основі попередньо визначених шаблонів. Такий механізм дозволяє поєднувати різні джерела інформації та формувати релевантні відповіді відповідно до контексту запиту користувача.

Підсистема доступу до даних відповідає за зберігання та отримання інформації, необхідної для роботи чат-бота. Вона може включати базу знань, історію діалогів, профілі користувачів або інші інформаційні ресурси. Розділення логіки обробки запитів і рівня зберігання даних дозволяє підвищити продуктивність системи та забезпечити більш ефективне управління інформаційними ресурсами. Крім того, архітектура чат-бота часто передбачає інтеграцію з зовнішніми інформаційними системами через API, що дає змогу отримувати дані з корпоративних сервісів або інших програмних платформ [3].

У сучасних дослідженнях також розглядаються архітектурні моделі, у яких функції чат-бота розподіляються між кількома взаємодіючими агентами. У такій структурі окремі компоненти відповідають за аналіз запиту, пошук інформації, перевірку результатів або формування відповіді. Мультиагентний підхід підвищує гнучкість системи та дозволяє ефективніше організувати взаємодію між різними функціональними модулями.

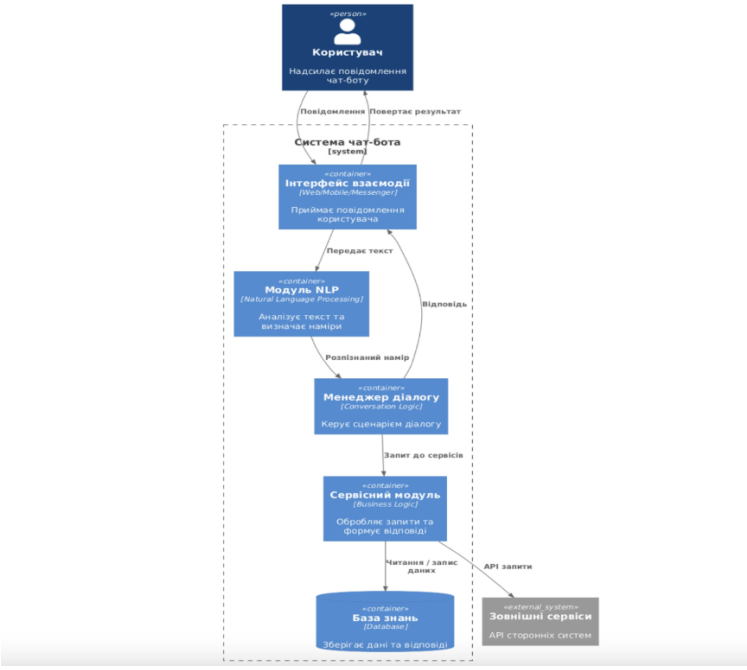


Рисунок 1 – Архітектура програмної системи чат-бота

Таким чином, архітектура сучасного чат-бота формується як модульна програмна структура, що поєднує інтерфейсні компоненти, механізми обробки природної мови, систему керування діалогом та підсистеми доступу до даних і зовнішніх сервісів. Використання таких архітектурних принципів забезпечує масштабованість, спрощує розвиток функціоналу та дозволяє створювати ефективні діалогові системи для веб-сервісів, мобільних застосунків і корпоративних інформаційних платформ [1,2].

Список використаних джерел

1. Ciesla R. The Book of Chatbots: From ELIZA to ChatGPT. Springer Nature, 2024. 159 c.
2. Skuridin A., Wynn M. Chatbot Design and Implementation: Towards an Operational Model for Chatbots. Mdpi. 17.04.2024. URL:https://www.mdpi.com/2078-2489/15/4/226 (дата звернення: 15.03.2026).
3. Aleedy M., Atwell E., Meshoul S. A Multi-Agent Chatbot Architecture for AI-Driven Language Learning. Mdpi. 01.10.2025. URL: https://www.mdpi.com/2076-3417/15/19/10634 (дата звернення: 15.03.2026).

УДК 004.738.5

РОЗРОБКА ФРОНТЕНДУ ЗА ДОПОМОГОЮ РІЗНИХ ФРЕЙМВОРКІВ: ПЕРЕВАГИ, НЕДОЛІКИ ТА ЗАВДАННЯ

Стеценко Я. І.
yanastetforcomp@gmail.com
Черкаський державний фаховий бізнес-коледж
Подорошко Д. І.
м. Черкаси, Україна

Сучасна розробка фронтенду є фундаментальною складовою створення прикладних рішень, що відповідають високим вимогам користувачів щодо швидкодії, масштабованості та безпеки. Еволюція вебтехнологій призвела до домінування односторінкових додатків (SPA), які забезпечують миттєву реакцію

без перезавантаження сторінок. Ці рішення базуються на JavaScript та TypeScript. Вибір інструменту є стратегічним рішенням, що впливає на життєвий цикл продукту. Індустрія сформувала чітку трійку лідерів: React, Vue.js та Angular, кожен з яких має унікальну архітектуру та специфічні сфери застосування, які потребують детального технічного аналізу.

Технологія React, бібліотека для створення користувацьких інтерфейсів, підтримується Meta. Її ключова інновація – використання Віртуального DOM (Virtual DOM). React обчислює різницю між поточним та новим станом (алгоритм узгодження) і точково оновлює лише необхідні елементи реального DOM, що сприяє підвищенню продуктивності у складних інтерфейсах.

Суворий компонентний підхід дозволяє розбивати інтерфейси на дрібні, незалежні та придатні для повторного використання блоки коду. Це значно спрощує командну розробку, тестування та рефакторинг. Величезна спільнота та широка екосистема роблять React універсальним інструментом.

Серед недоліків: високий поріг входження через відсутність суворої стандартизації архітектури «з коробки» (розробникам доводиться самостійно обирати інструменти для маршрутизації та управління станом, наприклад, Redux або Zustand). Часті оновлення парадигм, зокрема перехід до функціональних компонентів з хуками, вимагають постійного навчання та адаптації існуючого коду. React використовують для створення інтерактивних дашбордів, розгалужених платформ електронної комерції, соціальних мереж та стрімінгових сервісів. Наприклад, під час розробки проєкту MusicHub компонентний підхід React дозволив команді ефективно реалізувати динамічний каталог нотного матеріалу з миттєвим пошуком та створити безперебійний процес оформлення підписки, забезпечивши високу стабільність платформи. [1]

Vue.js – це прогресивний фреймворк, розроблений для об'єднання найкращих архітектурних рішень та створення максимально зрозумілого та гнучкого продукту. Він характеризується надзвичайно адаптивною системою інтеграції, дозволяючи використовувати його як легку бібліотеку для

інтерактивності на окремих сторінках, або як повноцінний фреймворк для складних корпоративних додатків (із Vue Router та Pinia).

Найсильнішою стороною є реактивна система збору залежностей, яка працює практично непомітно для розробника, автоматично відстежуючи зміни в даних та миттєво оновлюючи інтерфейс. Розробники високо оцінюють парадигму однофайлових компонентів (JavaScript, HTML, CSS в єдиному файлі), що підвищує візуальну зрозумілість структури проєкту. Слабкою стороною традиційно виділяють дещо меншу популярність у великому корпоративному секторі порівняно з конкурентами, хоча ситуація швидко змінюється. Vue.js часто застосовується для розробки фронтенд-інтерфейсів сучасних фінансових сервісів та цифрових гаманців. Висока швидкість рендерингу, легкість у підтримці коду та зручність управління складним фінансовим станом дозволяють створювати надійні клієнтські рішення, які витримують значні навантаження. [2]

Angular – це комплексний фреймворк, який розробляється та підтримується корпорацією Google. На відміну від React та Vue, Angular пропонує філософію повної комплектації, містячи вбудовані стандартизовані інструменти для всіх можливих завдань веброзробки: від маршрутизації та роботи з формами до HTTP-запитів та глибокого модульного тестування. Базовою мовою програмування є TypeScript, що забезпечує сувору типізацію коду, використання об'єктно-орієнтованих патернів та виявлення архітектурних помилок ще на етапі компіляції.

Архітектура Angular суворо базується на концепціях модульності, сервісів та впровадження залежностей (Dependency Injection), що робить його ідеальним вибором для великих інженерних команд, де критично важливо дотримуватися єдиного суворого стилю написання коду. Angular характеризується складнішим порогом входження, оскільки для продуктивної роботи розробникам необхідно глибоко опанувати концепції реактивного програмування за допомогою бібліотеки RxJS. Монолітність та великий початковий розмір кінцевого пакета часто роблять його технічно невиправданим для невеликих проєктів. Основна

ніша застосування Angular – розробка масштабних корпоративних систем управління, складних CRM та банківських порталів, що мають високі вимоги до стабільності, прогнозованості та довготривалої підтримки. [3]

Окрім вибору безпосередньо фреймворку, сучасна розробка фронтенду вимагає врахування архітектури безпеки, зокрема, впровадження концепції нульової довіри (Zero Trust) на рівні клієнтського додатка. Фронтенд-розробники повинні забезпечувати надійне зберігання авторизаційних даних, правильне налаштування політик спільного використання ресурсів та безпечну обробку токенів сесій. Хоча сучасні версії фреймворків автоматично екранують дані, мінімізуючи ризики міжсайтового скриптингу (XSS), розвиток технологій штучного інтелекту стимулює появу нових загроз. Тому вибір сучасного та регулярно оновлюваного фреймворку є критичною необхідністю для забезпечення стабільності інформаційних систем та захисту даних користувачів.

Детальний технічний аналіз сучасного інструментарію веброзробки підтверджує, що вибір фронтенд-технології залежить від специфіки конкретного продукту, вимог до його довгострокового масштабування та наявного досвіду команди:

- React залишається найбільш універсальним та гнучким вибором для більшості динамічних платформ, де ключову роль відіграє насичена взаємодія з користувачем та швидкість відмальовування компонентів.
- Vue.js приваблює структурною елегантністю, відносно низьким порогом входження та простотою поступової інтеграції, що робить його затребуваним при створенні цифрових гаманців та легких комерційних сервісів.
- Angular міцно широко застосовується для побудови корпоративних інформаційних систем найвищої складності, де пріоритетами виступають сувора типізація, жорстка прогнозованість архітектури та наявність єдиного стандартизованого набору інструментів.

Практика розробки підтверджує, що грамотний та обґрунтований вибір технологічного стека дозволяє не лише значно оптимізувати процеси

програмування, але й гарантувати створення надійного, стійкого до сучасних кіберзагроз та максимально зручного інтерфейсу, що відповідає найвищим стандартам якості програмного забезпечення.

Список використаних джерел:

1. React – A JavaScript library for building user interfaces. Official Documentation. URL: https://react.dev/.
2. MDN Web Docs: Front-end web developer. Mozilla Corporation. URL: https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer
3. OWASP Top 10 Client-Side Security Risks. Open Worldwide Application Security Project. URL: https://owasp.org/www-project-top-10-client-side-security-risks/.
4. Web Security Guidelines. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/Security.
5. Osmani A. Learning JavaScript Design Patterns: A JavaScript and React Developer's Guide. 2nd Edition. O'Reilly Media, 2023. 280 p.

УДК 004.41, 004.75

ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ РОЗПОДІЛЕНИХ СИСТЕМ КОНТРОЛЮ ВЕРСІЙ

Базюк В.Р
makamagol@gmail.com
Черкаський державний фаховий бізнес-коледж
Марченко С. В.
Черкаси, Україна

Розподілені системи контролю версій (Distributed Version Control Systems, DVCS) є критично важливим класом інженерного програмного забезпечення, що поєднує збереження історії змін, асинхронну реплікацію, цілісність даних і підтримку паралельної розробки. Сучасні системи цього класу, зокрема Git, Mercurial, Fossil і Darcs, реалізують подібні базові функції, але істотно

відрізняються за моделлю збереження об’єктів, організацією історії, механізмами синхронізації та оптимізаціями доступу до великих репозиторіїв. Оскільки саме ці внутрішні архітектурні рішення визначають масштабованість, швидкодію та надійність DVCS, аналіз їхніх функціональних компонентів і програмних реалізацій є актуальним інженерним завданням.

Здійснити критичний аналіз основних функціональних компонентів DVCS, дослідити варіації їх програмної реалізації в поширених системах контролю версій та визначити ключові архітектурні компроміси між продуктивністю, ефективністю збереження даних, зручністю синхронізації й складністю супроводу.

Git офіційно позиціонується як швидка, масштабована розподілена система керування версіями з доступом як до високорівневих, так і до внутрішніх механізмів. Порівняльна характеристика реалізацій DVCS наведена в табл. 1.

У сценаріях роботи з дуже великими репозиторіями (монорепозиторіями) кращі результати демонструє архітектура системи Git. Хороші результати досягаються поєднанням моделі збереження знімків стану (snapshot-model) із багаторівневою системою оптимізацій доступу до даних. Часткове клонування (partial clone) дає змогу завантажувати лише підмножину об’єктів, необхідних для поточного робочого контексту, а індекси досяжності об’єктів у вигляді бітових карт (bitmap indexes) і повторне використання пакетів об’єктів (pack reuse) суттєво скорочують час виконання операцій отримання змін (fetch) і клонування репозиторію (clone) [1; 7; 9].

Фактично це означає, що продуктивність Git забезпечується не лише базовою графовою моделлю історії у вигляді орієнтованого ациклічного графа (directed acyclic graph, DAG) і контентно-адресованим сховищем (content-addressable storage), а й використанням спеціалізованих індексних структур, які дозволяють замінити повний обхід історії попередньо обчисленими метаданими. Недоліком такого підходу є зростання складності супроводу репозиторію, оскільки для підтримання ефективності необхідно виконувати процедури перепакування (repack) об’єктів і збирання «сміття».

Таблиця 1 – Порівняльна характеристика реалізацій розподілених СКВ

№	Функціональний компонент	Git	Mercurial	Fossil	Darcs
1	Модель збереження даних	Збереження знімків стану з подальшим пакуванням об’єктів (snapshot, packfiles) [1; 2]	Журнали ревізій із збереженням різниць між версіями (revlog, generaldelta) [3]	Збереження артефактів у вбудованій базі даних (SQLite) зі стисканням різниць [4]	Збереження змін як окремих патчів [5]
2	Модель історії змін	Орієнтований ациклічний граф комітів з додатковими індексами [6]	Граф змін (changeset DAG) [3]	Граф артефактів репозиторію (Artifact DAG) [4]	Алгебра змін із можливістю перестановки патчів (Patch algebra) [5]
3	Механізми реплікації	Інтелектуальний протокол синхронізації з вибірковим отриманням даних [7]	Передавання груп змін і бандлів історії використання [8]	Синхронізація через мережеві служби HTTP або SSH [4]	Обмін множинами патчів [5]
4	Алгоритми узгодження змін	Рекурсивне тристороннє злиття [1]	Тристороннє злиття (Three-way merge) [3]	Злиття на рівні файлів [4]	Формальна комутація змін (Patch commutation) [5]
5	Оптимізація продуктивності	Індекси досяжності, багатопакетні індекси, повторне використання пакетів [9]	Ланцюги різниць і оптимізоване стискання історії (Delta chains) [3]	Індексація на рівні бази даних [4]	Формалізоване впорядкування змін (Algebraic reasoning) [5]

У випадках, коли історія розвитку проєкту характеризується значною кількістю розгалужень і частими операціями злиття, архітектура системи Mercurial може демонструвати кращі результати. Це пов’язано з використанням структури журналів ревізій (revlog) і формату загальних дельт (generaldelta), який дозволяє будувати ланцюги різниць (delta chains) між довільними версіями, а не лише між сусідніми [3]. Завдяки цьому досягається кращий коефіцієнт стискання історії змін і більш передбачувана продуктивність операцій запису нових змін. Модель append-only додавання зменшує необхідність модифікації вже збережених структур і тим самим підвищує надійність сховища. Водночас реконструкція повного стану репозиторію може вимагати обробки довших

послідовностей різниць, що збільшує затримку операції відновлення робочої директорії (checkout).

Для невеликих команд або автономних середовищ розробки ефективною може бути архітектура системи Fossil. Використання вбудованої системи керування базами даних SQLite як єдиного сховища дає змогу зберігати всі артефакти, метадані та індекси у вигляді одного файлу репозиторію [4]. Кращі результати тут досягаються за рахунок високого рівня інтеграції: система контролю версій, засоби відстеження помилок і вебінтерфейс адміністрування реалізовані в межах однієї програмної платформи. У результаті це зменшує залежність від зовнішньої інфраструктури та спрощує резервне копіювання. Разом із тим така централізація створює потенційні вузькі місця масштабування при інтенсивній паралельній роботі великої кількості розробників.

У дослідницьких сценаріях або спеціалізованих системах, де важливо формально описувати залежності між змінами, концептуальні переваги може мати архітектура системи Darcs. Вона базується на теорії патчів (patch theory), яка розглядає історію як множину змін, що можуть переставлятися за певних умов (commutation) [5]. Кращі результати в цьому випадку пов’язані з можливістю більш строгого логічного узгодження змін і потенційним зменшенням кількості конфліктів. Проте на практиці це супроводжується зростанням алгоритмічної складності аналізу залежностей і зниженням продуктивності при роботі з великими історіями змін, що обмежує застосування такого підходу у промислових масштабах.

У роботі запропоновано архітектуру вебплатформи проведення онлайн-вікторин у режимі реального часу. Її побудовано як modular monolith із чітким відокремленням REST-взаємодії, каналу передавання подій у режимі реального часу та модуля керування станом гри. Критичний аналіз показав, що для системи такого класу обраний підхід є обґрунтованішим за мікросервісну декомпозицію на ранньому етапі, оскільки дозволяє зменшити латентність, уникнути надмірної інфраструктурної складності та забезпечити узгоджений стан ігрової сесії.

Список використаних джерел:

1. Chacon S., Straub B. Pro Git. 2nd ed. New York : Apress, 2014. 456 p.
2. Git internals – Git objects . URL: https://git-scm.com/book/en/v2/Git-Internals-Git-Objects (дата звернення: 16.03.2026).
3. O'Sullivan B. Mercurial: the definitive guide . URL: https://book.mercurial-scm.org/read/concepts.html (дата звернення: 16.03.2026).
4. Fossil SCM technical overview. URL: https://fossil-scm.org/home/doc/trunk/www/tech_overview.wiki (дата звернення: 16.03.2026).
5. Darcs theory . URL: https://darcs.net/Theory (дата звернення: 16.03.2026).
6. Git commit-graph documentation . URL: https://git-scm.com/docs/commit-graph (дата звернення: 16.03.2026).
7. Git partial clone . URL: https://git-scm.com/docs/partial-clone (дата звернення: 16.03.2026).
8. Mercurial wire protocol. URL: https://www.mercurial-scm.org/wiki/WireProtocol (дата звернення: 16.03.2026).
9. Git pack objects . URL: https://git-scm.com/docs/git-pack-objects (дата звернення: 16.03.2026).

УДК 004.4:004.2

РЕТРО-ТЕХНОЛОГІЇ: ЧИ МОЖЛИВО ВІДТВОРИТИ ПРОГРАМУВАННЯ ДЛЯ ЗАСТАРІЛИХ ПЛАТФОРМ

Собчук Є. О.
yehor.sobchuk@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Актуальність роботи зумовлена широким використанням застарілих інформаційних технологій у критично важливих галузях, таких як фінансовий сектор, державне управління, енергетика та охорона здоров'я. Незважаючи на стрімкий розвиток сучасних IT-рішень, значна частина інформаційних систем

продовжує функціонувати на базі застарілих програмних платформ, що обумовлено їх надійністю, високою вартістю модернізації та складністю міграції.

Водночас використання таких технологій супроводжується низкою проблем, зокрема підвищеними ризиками кібербезпеки, обмеженою сумісністю з сучасними системами та дефіцитом кваліфікованих фахівців для їх підтримки. У цих умовах особливої актуальності набуває питання оцінювання доцільності збереження, модернізації або заміни застарілих технологій, а також можливості відтворення програмування для таких платформ.

Застарілі технології (obsolete technologies) визначаються як програмні та апаратні засоби, що втратили масове застосування або офіційну підтримку виробників, однак продовжують функціонувати в окремих галузях. Відповідно до сучасних досліджень, їх існування є природним наслідком швидкого розвитку інформаційних технологій та обмеженої можливості повної заміни складних інформаційних систем [1].

Життєвий цикл технологій передбачає етапи впровадження, активного використання та поступового витіснення більш ефективними рішеннями. Проте, на відміну від класичної моделі, у сфері програмного забезпечення спостерігається явище «технологічної інерції», коли застарілі системи продовжують експлуатуватися через високу вартість їх заміни або критичність для бізнес-процесів [2].

Критичний аналіз показує, що твердження про повну неактуальність старих технологій є спрощеним. Наприклад, мова програмування COBOL, створена у 1959 році, досі широко використовується у фінансовому секторі та державних установах. За оцінками аналітиків, значна частина банківських транзакцій у світі обробляється системами, написаними на COBOL [3]. Це свідчить про високу стабільність та надійність таких систем, але водночас створює проблему дефіциту фахівців.

З іншого боку, використання застарілих технологій має суттєві недоліки. Основними з них є:

- відсутність регулярних оновлень безпеки;
- обмежена сумісність із сучасними стандартами;
- складність інтеграції з новими системами.

Ці фактори значно підвищують ризики кіберзагроз. Зокрема, відсутність оновлень безпеки робить системи вразливими до відомих експлойтів, що підтверджується звітами міжнародних організацій у сфері кібербезпеки [4].

Водночас слід зауважити, що не всі сучасні технології є принципово новими. Більшість концепцій програмування (алгоритми, структури даних, принципи управління пам’яттю) були сформовані ще в середині XX століття. Сучасні мови програмування переважно розвивають ці підходи, додаючи нові рівні абстракції та покращуючи зручність використання [5].

Критичний аналіз також показує, що проблема застарілих технологій часто полягає не стільки в їх віці, скільки в якості реалізації програмних систем. Низька якість коду, відсутність документації та застаріла архітектура є більш значущими факторами ризику, ніж сама технологія [2].

У сучасних умовах доцільно застосовувати комбінований підхід, що передбачає:

- поступову модернізацію існуючих систем;
- використання методів міграції даних;
- впровадження сучасних засобів захисту інформації;
- підготовку фахівців для підтримки критично важливих застарілих систем.

Таким чином, ретро-технології не слід розглядати виключно як застарілий та неактуальний інструмент. У багатьох випадках вони залишаються важливою складовою інформаційної інфраструктури, хоча й потребують модернізації та додаткових заходів безпеки.

Застарілі технології є невід’ємною частиною сучасного IT-середовища. Їх використання обумовлене економічними, технічними та організаційними чинниками. Незважаючи на наявні ризики, повна відмова від таких систем є часто недоцільною. Оптимальним підходом є їх поетапна модернізація та інтеграція із сучасними технологіями.

Список використаних джерел:

1. Laudon K. C., Laudon J. P. Management Information Systems. New York: Pearson, 2021. 560 p.
2. Sommerville I. Software Engineering. 10th ed. Boston: Pearson, 2021. 816 p.
3. IBM Corporation. COBOL Systems and Business Applications. URL: https://www.ibm.com (дата звернення: 15.03.2026).
4. ENISA. Threat Landscape 2023. Luxembourg: Publications Office of the European Union, 2023. 120 p.
5. Sebesta R. W. Concepts of Programming Languages. 12th ed. Boston: Pearson, 2022. 720 p.

УДК 004.42

МЕХАНІЗМИ УНАОЧНЕННЯ АЛГОРИТМІВ ТА ЕФЕКТИВНІ СПОСОБИ ЇХ ВІЗУАЛІЗАЦІЇ ЗАСОБАМИ ВЕБТЕХНОЛОГІЙ

Близнюк В. П.
iterbleiker1@gmail.com
Черкаський державний фаховий бізнес-коледж
Марченко С. В.
м.Черкаси, Україна

Унаочнення алгоритмів використовується як засіб підтримки розуміння обчислювальних процесів, налагодження програм і дослідження поведінки структур даних. На відміну від статичних описів алгоритмів, інтерактивні візуальні представлення відображають зміну станів під час виконання, що може бути корисним як у навчальних, так і в інженерних задачах. Сучасні вебзастосунки дають змогу поєднати в одному середовищі код, візуальну модель, історію виконання та керування сценаріями взаємодії. Автори [1] вказують, що корпус емпіричних досліджень у цій галузі все ще є відносно обмеженим, тобто наявність відкритих питань щодо того, які саме механізми унаочнення є справді результативними.

Проаналізувати сучасні підходи до унаочнення алгоритмів і визначити механізми візуалізації, які в умовах вебсередовища мають найбільшу практичну цінність для розуміння, налагодження та дослідження поведінки алгоритмів.

У сучасних оглядових роботах візуалізацію програм розглядають уже не як окремий навчальний прийом, а як багатошаровий засіб аналізу програмної поведінки. Зокрема, в огляді [2] подано класифікацію за етапами виконання: до запуску (для аналізу структури коду й потенційних помилок), під час виконання (для розуміння алгоритму, налагодження та моніторингу станів), після виконання (для оцінювання продуктивності, оптимізації та виявлення аномалій). Така рамка є важливою, оскільки зміщує фокус із привабливої анімації на інструментальну корисність візуального подання.

Лише факт наявності візуалізації ще не гарантує кращого розуміння алгоритму. Найбільш переконливі результати з'являються там, де візуальне подання поєднане з керуванням виконанням, відображенням проміжних станів і можливістю перевіряти власні припущення користувача.

Перший механізм – покрокове трасування з явним поданням станів. Його суть полягає в фіксації стану даних, керівних переходів і наслідків кожної операції. Такий підхід зберігає значення для класичних алгоритмів, але в сучасних системах він дедалі частіше доповнюється історією виконання та поверненням до попередніх кроків. Це переводить візуалізацію з демонстраційного режиму в режим дослідження. Огляд [2] прямо відносить такі сценарії до середньої фази виконання, де візуалізація підтримує розуміння, налагодження і моніторинг.

Другий механізм – живе програмування (live programming), тобто безперервне оновлення візуального подання під час редагування та виконання коду. У праці [3] на прикладі середовища Algot досліджено, наскільки такий режим допомагає у розумінні програм. Автори показали, що ефект не є універсальним, проте для задач, пов'язаних із матрицями та деревами, зафіксовано кращі результати; крім того, учасники дещо вище оцінили зручність

і задоволення від роботи із середовищем. Цей кейс демонструє вибіркову ефективність такого підходу залежно від типу задачі.

Автори [4] дійшли висновку, що безперервне відображення поточних значень може зменшувати когнітивне навантаження під час валідації та послаблювати як надмірну довіру, так і надмірну недовіру до підказок ШІ. Для теми унаочнення алгоритмів це показово, оскільки візуалізація тут працює не лише як навчальний засіб, а як інструмент інженерної перевірки рішень.

Третій механізм – візуалізація власного коду користувача, а не лише заздалегідь підготовлених прикладів. У системі PVLS блок-схеми та подання процесу виконання динамічно генерувалися зі студентського коду. За результатами дослідження [5] зафіксовано статистично значуще покращення загального результату та власне програмної успішності, але не отримано значущого приросту для розуміння процесу виконання як окремого показника. Отже, такий механізм виглядає перспективним для підтримки написання й налагодження коду, проте не варто беззастережно ототожнювати його з глибшим алгоритмічним розумінням.

Четвертий механізм – поєднання візуалізації з діалоговим або пояснювальним супроводом. У платформі VisualCodeMOOC динамічні візуалізації інтегровано з розмовним агентом VisualCodeChat і персоналізованим зворотним зв’язком. Автори [6] позиціонують систему як цілісне середовище активного навчання, де візуалізація не ізольована, а працює разом з поясненням, вправами й навігацією за завданням.

Окремої уваги заслуговує розширення цієї тематики на складніші алгоритмічні системи. Наприклад, у контексті машинного навчання потреба в унаочненні виникає не лише після побудови моделі, а й на різних етапах конвеєра аналізу для розуміння, діагностики й удосконалення моделей [7].

Разом із тим, новітні підходи не знімають низку обмежень. По-перше, інтерактивність ускладнює інтерфейс і створює ризик когнітивного перевантаження. По-друге, позитивний ефект часто виявляється локальним для певного типу задач або певної аудиторії, як це видно з кейсів Algot [3] і PVLS [5].

По-третє, навіть у найбільш сучасних роботах значна частина рішень залишається орієнтованою на навчальні сценарії, тоді як інтеграція з реальними середовищами розробки і системами налагодження ще не стала загальним стандартом.

У вебсередовищі реалізація таких механізмів унаочнення пов’язана з використанням клієнтських технологій відображення та керування станом виконання. Зокрема, застосовуються векторні графічні формати SVG і Canvas-відтворення для динамічного оновлення структури даних, а також механізми реактивного програмування, що забезпечують синхронізацію між змінами стану алгоритму і візуальним поданням. Для організації покрокового виконання використовуються подієві моделі браузера, асинхронні виклики та часові контролери анімацій. Крім того, обчислювально інтенсивні фрагменти алгоритмів можуть переноситися у WebAssembly-модулі, що дає змогу поєднати інтерактивність вебінтерфейсу з достатньою продуктивністю виконання.

Сучасний стан досліджень дає підстави розглядати унаочнення алгоритмів не як сукупність окремих анімацій, а як клас інтерактивних механізмів аналізу виконання. Для вебсередовища найбільш перспективними виявляються: покрокове трасування зі збереженням історії станів, живе оновлення подання під час роботи з кодом, побудова візуального подання на основі власного коду користувача, а також поєднання візуалізації з пояснювальними та діалоговими засобами. Водночас емпірична база ще не є достатньо щільною, а результати окремих кейсів свідчать, що ефективність залежить від типу задачі, форми взаємодії та цілей застосування. Саме тому подальші дослідження доцільно спрямувати на порівняння механізмів унаочнення в однакових умовах, а також на їх інтеграцію з інструментами налагодження та аналізу програмної поведінки.

Список використаних джерел:

1. Liu J., Poulsen S., Goodwin E., Chen H., Williams G., Gertner Y., Franklin D. Teaching Algorithm Design: A Literature Review // ACM Transactions on

Computing Education. 2025. Vol. 25. Issue 2. Article No. 17 P. 1-20. DOI: 10.1145/3727987.

2. Zhang W., Wen Z., Pan J., Chen W. Visualization for Computer Program: A Survey // Journal of Computer-Aided Design & Computer Graphics. 2023. Vol. 35, No. 8. P. 1139–1149. DOI: 10.3724/SP.J.1089.2023.19893.
3. Graf O., Thorgeirsson S., Su Z. Assessing Live Programming for Program Comprehension // Proceedings of the 2024 on Innovation and Technology in Computer Science Education. Vol. 1. 2024. P. 520–526. DOI: 10.1145/3649217.3653547.
4. Ferdowsi K., Huang R., James M. B., Polikarpova N., Lerner S. Validating AI-Generated Code with Live Programming // Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems. 2024. Article No. 143. P. 1–8. DOI: 10.1145/3613904.3642495.
5. Lai C.-H., Lin P.-W., You S.-H. Development and Evaluation of a Dynamic Code Visualization System for C Programming Education: The PVLS Approach // Social Sciences & Humanities Open. 2025. Vol. 12. Art. 101962. DOI: 10.1016/j.ssaho.2025.101962.
6. Li M., Wang D., Purwanto E., Selig T., Zhang Q., Liang H.-N. VisualCodeMOOC: A Course Platform for Algorithms and Data Structures Integrating a Conversational Agent for Enhanced Learning Through Dynamic Visualizations // SoftwareX. 2025. Vol. 30. Art. 102072. DOI: 10.1016/j.softx.2025.102072.
7. Wang J., Liu S., Zhang W. Visual Analytics for Machine Learning: A Data Perspective Survey // IEEE Transactions on Visualization and Computer Graphics. 2024. Vol. 30, No. 12. P. 7637–7656. DOI: 10.1109/TVCG.2024.3357065.

УДК 004.451.9:004.42

ВИКОРИСТАННЯ СЦЕНАРІЇВ POWERSHELL ДЛЯ НАЛАШТУВАННЯ РОБОЧИХ СТАНЦІЙ

Бровко Д. Д.
my4enie777@gmail.com
Черкаський державний фаховий бізнес коледж
Фальченко Н. Г.
м.Черкаси, Україна

У сучасній системній інженерії та управлінні IT-інфраструктурою особливої ваги набуває концепція «інфраструктура як код» (Infrastructure as Code). Автоматизація розгортання та налаштування робочих станцій дозволяє суттєво скоротити час на введення обладнання в експлуатацію, мінімізувати людський фактор та забезпечити ідемпотентність – стан, при якому повторне виконання скрипта приводить до одного й того самого прогнозованого результату. Використання сценаріїв PowerShell є одним із найбільш ефективних методів керування середовищем Windows завдяки глибокій інтеграції з операційною системою та доступу до об’єктної моделі .NET [1].

Основні завдання розробки включають: аналіз існуючих методів розгортання (MDT, SCCM, ручне налаштування), проєктування модульної архітектури скриптів, реалізацію механізмів автоматичного встановлення прикладного ПЗ, конфігурування параметрів безпеки та мережевих інтерфейсів, а також створення системи звітності про стан завершення процесів [2].

Система реалізована на базі мови сценаріїв PowerShell версії 5.1/7.x. Взаємодія з компонентами ОС базується на використанні WMI (Windows Management Instrumentation) та CIM-сесій, що дозволяє отримувати вичерпну інформацію про апаратне забезпечення та керувати системними службами. Головною перевагою такого підходу є відсутність потреби у сторонніх агентах (agentless-підхід) та можливість гнучкого налаштування реєстру, профілів користувачів і системних політик. Для управління пакунками ПЗ інтегровано використання менеджерів пакетів, таких як Winget та Chocolatey, що забезпечує

автоматичне завантаження та «тихе» встановлення актуальних версій програм [3].

Особлива увага приділяється забезпеченню безпеки виконання сценаріїв. Реалізовано перевірку політик виконання (Execution Policy), обробку виняткових ситуацій через блоки Try-Catch-Finally та логування кожного етапу розгортання у текстові файли або події Windows (Event Log). Це критично важливо для діагностики помилок у великих корпоративних мережах. Крім того, розроблено механізм ідентифікації обладнання, що дозволяє скрипту адаптивно встановлювати специфічні драйвери залежно від моделі робочої станції [4].

Функціонал розробленого комплексу сценаріїв включає:

- модуль первинної ініціалізації ОС (регіональні стандарти, назва ПК);
- блок керування мережевими налаштуваннями та приєднанням до домену Active Directory;
- систему автоматичного встановлення базового та спеціалізованого ПЗ;
- модуль оптимізації продуктивності та вимкнення небажаних телеметричних служб;
- інтерфейс фінальної перевірки цілісності конфігурації. Усі компоненти побудовані за принципом модульності, що дозволяє легко додавати нові функції без зміни ядра системи [5].

Логіка роботи побудована на послідовному виконанні фаз: ініціалізація, конфігурування, верифікація. Для прискорення процесу впроваджено методи паралельних обчислень (PowerShell Jobs/Runspaces) при встановленні незалежних компонентів ПЗ. Для інформування системного адміністратора про хід роботи додано графічні прогрес-бари або консольну індикацію статусів.

Додатково реалізовано систему аудиту, яка після завершення роботи скриптів формує фінальний звіт (HTML або CSV). Це дозволяє миттєво оцінити, чи всі параметри було застосовано коректно, та зафіксувати час, витрачений на підготовку однієї станції. Енергоаудит та моніторинг ресурсів під час виконання скриптів підтвердили низьке навантаження на процесор та пам’ять.

Тестування проводиться на віртуальних машинах та фізичних робочих станціях з різними конфігураціями заліза. Результати показали, що автоматизований підхід скорочує час налаштування робочого місця на 70–85% порівняно з ручним методом. Пристрій демонстрував високу стабільність навіть за умов нестабільного мережевого з’єднання завдяки механізмам повторних запитів (Retry logic).

У перспективі планується інтеграція розроблених рішень із хмарними сервісами (Microsoft Intune/Azure Autopilot), додавання підтримки PowerShell Core для кросплатформного керування та розробка веб-інтерфейсу для моніторингу статусу розгортання в реальному часі.

Таким чином, розроблена система автоматизації на основі PowerShell є потужним інструментом для сучасного системного адміністратора, що поєднує гнучкість, масштабованість та високу надійність. Реалізовані технічні рішення дозволяють забезпечити уніфікацію робочих місць та значне підвищення ефективності ІТ-відділу.

Список використаних джерел:

1. Бернацький А. В. Автоматизація адміністрування Windows за допомогою PowerShell. Технічні науки та технології. 2021. № 2 (24). С. 112–118.
2. Шевченко О. М., Петренко В. І. Порівняльний аналіз інструментів розгортання робочих станцій у корпоративних мережах. Системи обробки інформації. 2023. Вип. 3 (172). С. 45–52.
3. Коваленко С. С. Використання менеджерів пакетів для автоматизації встановлення ПЗ. Комп’ютерно-інтегровані технології. 2022. Т. 48. С. 89–94.
4. Мельник Д. Р. Забезпечення безпеки та цілісності сценаріїв автоматизації в ОС Windows. Кібербезпека: освіта, наука, техніка. 2024. № 1 (21). С. 130–137.

5. Сидоренко В. А. Оптимізація IT-інфраструктури підприємства через впровадження методів Infrastructure as Code. Цифрова економіка та системи управління. 2024. Вип. 12. С. 15–22.

УДК 004.8:004.415

АРХІТЕКТУРНІ ПІДХОДИ ТА ІНЖЕНЕРНІ ПРАКТИКИ РОЗРОБКИ ДЕЦЕНТРАЛІЗОВАНИХ ВЕБ-ЗАСТОСУНКІВ НА БАЗІ БЛОКЧЕЙНУ

Скубій Є.В.
yevgeniyaskubiy@gmail.com
Черкаський державний фаховий бізнес-коледж
Дмитрюк В.В.
м.Черкаси, Україна

Розвиток Web 3.0 потребує створення децентралізованих застосунків (dApps), які забезпечують безпечну взаємодію з блокчейном без передачі приватних ключів. Перехід від моделі «клієнт-сервер» до децентралізованих систем вимагає нових підходів до архітектури, безпеки та оптимізації витрат (газ-комісій). Через складність управління станом транзакцій та специфіку UX, систематизація інженерних практик розробки dApps є критично важливою для забезпечення надійності та масштабованості сучасних вебсервісів.

Метою дослідження є аналіз сучасних архітектурних підходів та інженерних практик розробки децентралізованих вебзастосунків на базі блокчейну, систематизація вимог до їхнього проєктування та вивчення технологічних рішень для забезпечення безпеки, масштабованості й зручності використання [1].

Децентралізований застосунок базується на моделі «клієнт-блокчейн», що замінює класичну схему «клієнт-сервер». Архітектура включає три рівні: на рівні блокчейну смартконтракти (Solidity, Rust) реалізують логіку в детермінованому та невідворотному вигляді; проміжний рівень (JSON-RPC шлюзи та API) забезпечує зв'язок фронтенду з мережею; на рівні фронтенду інтерфейси взаємодіють із гаманцями (як MetaMask) для безпечного підписання транзакцій.

Така архітектура потребує переосмислення методів кешування та управління станом через високі затримки та вартість операцій у розподілених мережах [1, 4].

Безпека у блокчейн-застосунках досягається через комбінацію криптографічних механізмів, формальної верифікації та регулярного аудиту коду. Оскільки смартконтракти після розгортання зазвичай є незмінними (immutability), будь-які вразливості стають постійними, тому сучасна практика включає статичний аналіз коду (Slither, Mythril), динамічне тестування та зовнішні професійні аудити [4, 5]. На рівні фронтенду критичного значення набуває безпечна взаємодія з гаманцями: фішингові атаки та підробки сайтів потребують імплементації стандартів EIP (Ethereum Improvement Proposals), перевірки адрес отримувачів та попередження користувача перед підписанням. Використання перевірених бібліотек (Web3.js, ethers.js, viem) спрощує розробку, проте вимагає суворого дотримання код-рев'ю та моніторингу активності контрактів [1].

Сучасний технологічний стек розробки dApps зазвичай включає: смартконтракти (мова Solidity, фреймворки Hardhat або Foundry), фронтенд (React/Next.js, Vue), бібліотеки взаємодії з блокчейном (Web3.js, ethers.js, viem), індексування та запити до даних (The Graph) та засоби перевірки коду (Slither, Mythril).

Вибір інструментів впливає на швидкість розробки, безпеку та масштабованість. Hardhat забезпечує цілісне середовище розробки з вбудованою мережею для тестування та інтеграцією з популярними інструментами верифікації [2]. Foundry, написаний на Rust, пропонує швидший процес тестування та написання тестів на Solidity замість JavaScript [3]. Бібліотеки взаємодії з блокчейном спрощують підписання транзакцій та роботу з контрактами, тоді як OpenZeppelin надає перевірені реалізації стандартів безпеки для смартконтрактів [4].

Отже, провадження архітектурних підходів і інженерних практик у розробку децентралізованих вебзастосунків дозволяє забезпечити безпечну та надійну взаємодію користувачів із блокчейном. Використання сучасного

технологічного стеку, автоматизованих інструментів для тестування та візуалізації, а також систематизація процесів розробки сприяє підвищенню масштабованості, швидкості розробки та зменшенню впливу людських помилок. Це створює стабільну основу для розвитку Web 3.0 та впровадження нових децентралізованих сервісів.

Список використаних джерел:

1. Ethereum Development Documentation. Official Ethereum Protocol Documentation. URL: https://ethereum.org/en/developers/docs/ (дата звернення: 16.03.2025).
2. Hardhat Documentation. Ethereum development environment. URL: https://hardhat.org/ (дата звернення: 16.03.2025).
3. Foundry Book. Smart Contract Development Suite. URL: https://book.getfoundry.sh/ (дата звернення: 16.03.2025).
4. OpenZeppelin Contracts. Standard Smart Contract Library. URL: https://docs.openzeppelin.com/contracts/ (дата звернення: 16.03.2025).
5. Security Considerations for Smart Contracts / S. DeFi Security. Ethereum Foundation. URL: https://ethereum.org/en/developers/docs/smart-contracts/security/ (дата звернення: 16.03.2025).

УДК 004.42

УПРАВЛІННЯ З'ЄДНАННЯМИ З БАЗОЮ ДАНИХ У FLASK-ЗАСТОСУНКУ ДЛЯ ЗАПОБІГАННЯ БЛОКУВАННЮ SQLite

Валовий А.Є
angegter@gmail.com
Черкаський державний фаховий бізнес-коледж
Фальченко Н.Г.
м.Черкаси, Україна

Під час розробки веб-застосунку для автоматизованої перевірки алгоритмічних задач на Python виникла необхідність зберігати результати перевірок у базі даних. Як сховище було обрано SQLite через його простоту та

відсутність необхідності у налаштуванні окремого сервера. Проте в процесі тестування системи виявилась критична проблема, пов'язана з одночасним доступом кількох запитів до бази даних.

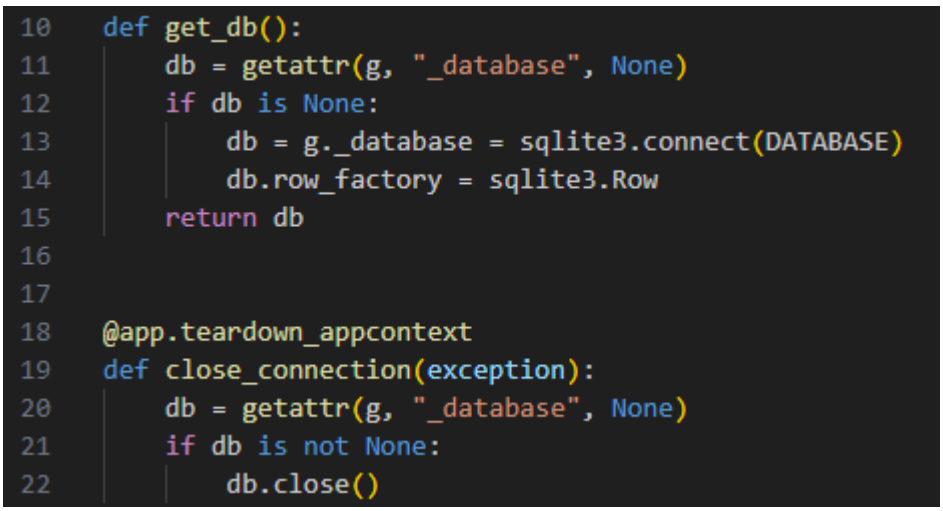
Flask, починаючи з версії 1.0, за замовчуванням обробляє HTTP-запити у багатопотоковому режимі (threaded=True). Це означає, що кілька запитів можуть оброблятися одночасно у різних потоках. SQLite за замовчуванням забороняє використання одного з'єднання у різних потоках через параметр check_same_thread=True.

При одночасному надходженні двох запитів на перевірку коду Flask намагався використати те саме з'єднання з БД у різних потоках. Це призводило до виключення sqlite3.OperationalError: database is locked, через яке другий запит аварійно завершувався, а результат перевірки не зберігався. Відтворити помилку вдалось лише при одночасному відкритті кількох вкладок браузера з однаковою задачею.

Перший підхід полягав у передачі параметра check_same_thread=False при створенні з'єднання. Це знімає обмеження SQLite на використання з'єднання між потоками, проте не вирішує проблему конкурентного запису – з'єднання залишається спільним, і при одночасних транзакціях блокування все одно виникає.

Коректним рішенням виявилось створення окремого з'єднання для кожного HTTP-запиту з використанням об'єкта g з модуля flask. Об'єкт g є локальним для поточного контексту запиту: з'єднання відкривається при першому зверненні до бази у функції get_db() і автоматично закривається після завершення запиту через декоратор teardown_appcontext. Таким чином кожен потік отримує власне ізольоване з'єднання.

Додатково було увімкнено режим WAL (Write-Ahead Logging) командою PRAGMA journal_mode=WAL. У цьому режимі операції читання не блокують запис і навпаки, що суттєво підвищує пропускну здатність при конкурентних запитах.



```
10  def get_db():
11      db = getattr(g, "_database", None)
12      if db is None:
13          db = g._database = sqlite3.connect(DATABASE)
14          db.row_factory = sqlite3.Row
15      return db
16
17
18  @app.teardown_appcontext
19  def close_connection(exception):
20      db = getattr(g, "_database", None)
21      if db is not None:
22          db.close()
```

Рисунок 1 – Реалізація управління з'єднаннями з базою даних у Flask

Після впровадження підходу з об'єктом g помилка database is locked зникла повністю. Система коректно опрацьовує одночасні запити на перевірку коду від різних користувачів. Кожне з'єднання існує рівно стільки, скільки триває HTTP-запит, що унеможливлює його захоплення іншим потоком. Увімкнення WAL-режиму додатково скоротило час відповіді при паралельному навантаженні.

Використання глобального з'єднання з SQLite у багатопотоковому Flask-застосунку призводить до блокувань і втрати даних. Правильним рішенням є створення з'єднання на рівні запиту через контекстний об'єкт g з автоматичним закриттям після відповіді. Такий підхід разом із WAL-режимом забезпечує стабільну роботу застосунку при одночасних зверненнях користувачів.

Список використаних джерел

1. Flask Documentation. The Application Context. URL: https://flask.palletsprojects.com/en/3.0.x/appcontext/ (дата звернення: 18.03.2026).
2. SQLite Documentation. WAL-mode. URL: https://sqlite.org/wal.html (дата звернення: 18.03.2026).
3. Grinberg M. Flask Web Development. 2nd ed. O'Reilly Media, 2018. 316 p.
4. Hettinger R. Python Cookbook: Recipes for Mastering Python 3. 3rd ed. O'Reilly Media, 2013. 706 p.

5. SQLite Documentation. SQLite Frequently Asked Questions. URL: https://sqlite.org/faq.html (дата звернення: 18.03.2026).

УДК 004.65

ІНТЕРАКТИВНА МАПА ЗМІН КЛІМАТУ НА ОСНОВІ ВЕЛИКИХ ДАНИХ

Сас Д. А.
dominikasas9110@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Зміна клімату є однією з найбільших проблем сьогодення [4]. Вона проявляється у підвищенні середньої температури повітря, зміні кількості опадів та у погодних явищах, таких як посухи, повені та урагани. Ці зміни впливають не лише на природу, але й на життя людей, економіку та сільське господарство.

Для дослідження кліматичних змін сьогодні активно використовуються інформаційні технології. Одним із найсучасніших підходів є використання великих даних (Big Data) [2]. Великі дані – це дуже великі обсяги інформації, які неможливо обробити звичайними способами.

Інтерактивна мапа змін клімату – це застосунок, який дозволяє збирати, аналізувати та показувати екологічні дані у зручному вигляді. Інтерактивна мапа може містити різні шари інформації. Наприклад, окремо можна переглянути рівень забруднення повітря, викиди вуглекислого газу, стан льодовиків або рівень моря. Користувач може переглядати інформацію про різні регіони світу, зміну температури та інші показники, може сам обирати, які саме дані його цікавлять, що робить додаток більш зручним і гнучким.

Основою такої системи є відкриті дані, які надаються міжнародними організаціями та науковими центрами [3]. Вони включають дані з метео-станцій, супутників, океанічних досліджень та інших джерел. Ці дані постійно оновлюються, тому мапа показує актуальну ситуацію.

Однією з переваг такого додатку є зручність використання. Навіть люди без спеціальних знань можуть легко зрозуміти, як змінюється клімат.

Також такі системи можуть використовуватися в освіті. Учні та студенти можуть вивчати кліматичні зміни на реальних даних, що робить навчання цікавішим.

Ще однією цікавою функцією є можливість переглянути, як змінювалася температура або рівень опадів за останні роки чи десятиліття.

На сьогодні вже існують подібні сервіси та застосунки. Наприклад, інтерактивні кліматичні мапи, які створюються міжнародними організаціями та науковими установами [1]. Вони показують зміну температури, рівень опадів та інші показники у різних частинах світу.

Отже, інтерактивна мапа змін клімату на основі великої кількості даних є важливим інструментом для аналізу екологічної ситуації. Вона допомагає краще зрозуміти глобальні проблеми та показати їх людям.

Список використаних джерел

1. Міністерство захисту довкілля та природних ресурсів України. URL: https://mepr.gov.ua (дата звернення: 31.03.2026).
2. Що таке великі дані URL: https://uk.wikipedia.org (дата звернення: 31.03.2026).
3. Національна академія наук України дослідження клімату. URL: https://www.nas.gov.ua (дата звернення: 31.03.2026).
4. Що таке зміна клімату? – Екологія життя. URL: https://ecolog-ua.com (дата звернення: 31.03.2026).

УДК 004.41

СИСТЕМА ОНЛАЙН-БРОНЮВАННЯ НЕРУХОМОСТІ З БАГАТОРІВНЕВОЮ МОДЕЛЛЮ ДОСТУПУ

Дулов О.А.
alex.dulov2006@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В.Ю.
м. Черкаси, Україна

Процеси цифровізації охоплюють дедалі більше сфер економіки, і ринок нерухомості не є винятком. Сучасні системи онлайн-бронювання мають забезпечувати не лише зручний інтерфейс для кінцевого користувача, а й гарантувати високий рівень безпеки персональних даних та цілісність фінансових транзакцій. Проблематика розробки таких систем полягає у необхідності розмежування прав доступу між різними категоріями користувачів (орендарями, орендодавцями та адміністраторами), що вимагає впровадження гнучких багаторівневих моделей авторизації.

Питання побудови розподілених систем бронювання та захисту даних у веб-середовищі розглядали багато вітчизняних та закордонних вчених. Проте швидкий розвиток хмарних технологій та нових методів аутентифікації потребує постійного оновлення підходів до проектування архітектури таких систем.

Метою роботи є проектування та обґрунтування архітектури системи онлайн-бронювання нерухомості з використанням багаторівневої моделі доступу – Role-Based Access Control (RBAC), що дозволить мінімізувати ризики несанкціонованого доступу до конфіденційної інформації.

Для реалізації системи обрано клієнт-серверну архітектуру. Backend-частина базується на REST API, що дозволяє в майбутньому легко масштабувати систему та підключати мобільні додатки. Основним елементом безпеки є впровадження багаторівневої моделі доступу.

Розглянемо детальніше функціональні рівні доступу:

1. Рівень гостя (неавторизований користувач): доступ лише до перегляду загального каталогу нерухомості, фільтрації за ціною та локацією.

2. Рівень клієнта (орендар): можливість створення профілю, управління кошиком бронювань, здійснення оплати та написання відгуків. Доступ до власних замовлень захищено індивідуальними токенами.
3. Рівень орендодавця (власник об'єкта): повний контроль над своїми оголошеннями, перегляд статистики зайнятості об'єктів, управління ціновою політикою та підтвердження заявок.
4. Рівень адміністратора: повний доступ до бази даних для модерації контенту, розв'язання конфліктних ситуацій та управління обліковими записами.

База даних системи розроблена з урахуванням нормалізації для забезпечення цілісності. Основні таблиці включають «Users», «Properties», «Bookings» та «Roles». Зв’язок між таблицями «Users» та «Roles» реалізовано за принципом «багато до багатьох», що дозволяє одному користувачу мати декілька ролей (наприклад, бути одночасно орендарем та орендодавцем).

Особлива увага приділена механізму «Race Condition» при одночасному бронюванні одного об’єкта двома користувачами. Для вирішення цієї проблеми застосовано песимістичне блокування записів у базі даних на рівні транзакцій, що унеможливлює паралельну зміну стану одного й того ж ресурсу. Це гарантує, що об’єкт змінить статус на «заброньовано» лише для того користувача, чий запит першим пройшов валідацію сервером.

Додатково розглянуто альтернативний підхід із використанням оптимістичного блокування, який базується на перевірці версій записів, проте він є менш надійним у сценаріях із високою конкуренцією запитів. Обраний підхід забезпечує цілісність даних та узгодженість транзакцій у реальному часі, що є критично важливим для систем онлайн-бронювання.

Безпека передачі даних забезпечується протоколом HTTPS та використанням JWT (JSON Web Tokens) для підтримки сесій. Кожен запит до захищених ресурсів перевіряється сервером на наявність дійсного токена та відповідність ролі користувача запитуваній дії.

Додатково реалізовано механізм оновлення токенів (refresh tokens), що підвищує безпеку та зручність роботи користувача без необхідності повторної аутентифікації. Для захисту системи від поширених веб-загроз застосовано перевірку вхідних даних та механізми протидії SQL-ін’єкціям, XSS та CSRF-атакам. Паролі користувачів зберігаються у вигляді хешів із використанням сучасних криптографічних алгоритмів.

Для підвищення надійності та контролю доступу передбачено ведення журналів подій (логування) та аудит дій користувачів, що дозволяє виявляти підозрілу активність і своєчасно реагувати на потенційні загрози.

Розроблена концепція системи онлайн-бронювання нерухомості з багаторівневою моделлю доступу дозволяє створити безпечне середовище для взаємодії власників та клієнтів. Використання моделі RBAC спрощує адміністрування системи та забезпечує масштабованість. Впровадження таких систем на вітчизняному ринку дозволить значно знизити витрати часу на пошук та оформлення оренди нерухомості, підвищуючи загальну прозорість ринку

Список використаних джерел:

1. Про авторське право і суміжні права : Закон України від 01.12.2022 № 2811-IX. URL: https://zakon.rada.gov.ua/laws/show/2811-20 (дата звернення: 17.03.2026).
2. Мазурок І. Є. Проектування інформаційних систем : навчальний посібник. Одеса : ОНЕУ, 2020. 184 с.
3. Жежнич П. І. Технології створення та ведення баз даних : підручник. Львів : Видавництво Львівської політехніки, 2018. 216 с.
4. Role-Based Access Control (RBAC). *NIST Computer Security Resource Center*. URL: https://csrc.nist.gov/projects/role-based-access-control (дата звернення: 17.03.2026).
5. Richardson L., Ruby S. RESTful Web Services: Service-Oriented Infrastructures for Distributed Applications. O'Reilly Media, 2017. 448 p.

6. Introduction to JSON Web Tokens. *Auth0 Docs*. URL: https://jwt.io/introduction/.
7. Спиваковський О. В., Щедролосєв Д. Є. Побудова безпечних розподілених веб-систем. Херсон : ХДУ, 2019. 150 с.

УДК 004.41

ПРОГРАМОТЕХНІКА ТА ПРОЄКТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ: АРХІТЕКТУРА КЛІЄНТСЬКОГО РІВНЯ ТА РЕАКТИВНІ ПАТЕРНИ

Кондратенко Є.С.
ekqwert12345@gmail.com
Черкаський державний фаховий бізнес-коледж
Подорошко Д.І.
м. Черкаси, Україна

У сучасній розробці програмного забезпечення клієнтська частина (front-end) вже не є лише оболонкою для відображення даних. Великі односторінкові застосунки (SPA) та корпоративні системи стикаються з подібними архітектурними ризиками, як і серверні рішення: неконтрольоване зростання зв'язаності модулів, каскадні збої та деградацію підтримуваності з часом. Браузерні застосунки функціонують автономно, реалізуючи складні обчислення, структурні патерни та бізнес-логіку на рівні, що відповідає серверній частині [1]. Проте специфіка середовища виконання, зокрема обмежені ресурси пристрою, відкритість коду та непередбачуваність мережі, унеможливлює пряме перенесення серверних рішень без адаптації. Архітектурні рішення, прийняті на ранніх етапах розробки клієнтського застосунку, визначають його довгострокову підтримуваність і стійкість до збоїв [2].

Компонентний підхід і сучасні практики у фреймворку Angular. Розробка інтерфейсів корпоративного рівня вимагає чіткої модульності та відокремлення бізнес-логіки від логіки користувацького інтерфейсу. Angular демонструє застосування інженерних методів у створенні клієнтських систем [3]. На відміну від легких бібліотек, Angular використовує статичну типізацію (TypeScript). Це

дозволяє виявляти помилки на етапі компіляції й зменшує їх кількість під час виконання [4]. Основою Angular є вбудований механізм впровадження залежностей (Dependency Injection, DI). Він забезпечує слабкий зв'язок між компонентами та сервісами. Це спрощує заміну реалізацій модулів, наприклад, для тестування, без зміни основного коду [3]. Компонентна архітектура інкапсулює HTML, CSS і логіку в окремі блоки, що сприяє повторному використанню коду [5].

На відміну від Angular, фреймворки React і Vue орієнтовані на гнучкість і мінімалізм. React пропонує декларативний підхід на основі компонентів і односторонній потік даних. Він не нав'язує суворої структури застосунку чи механізму DI, залишаючи архітектурні рішення на розсуд розробників. Vue акцентує на простоті й поступовому впровадженні. Він дозволяє інтегрувати сучасні концепції навіть у невеликі проєкти. Однак ні React, ні Vue не мають вбудованої системи DI чи обов’язкової статичної типізації. Тому великі команди часто впроваджують додаткові інструменти для забезпечення структурованості коду. Angular завдяки інтегрованій архітектурі є оптимальним вибором для масштабованих корпоративних систем. Тут важливими залишаються стандартизація та суворе дотримання інженерних практик.

Реактивне керування у клієнтських архітектурах є ключовим аспектом при застосуванні компонентного підходу. Управління станом у розподілених клієнтських архітектурах належить до найскладніших інженерних завдань, оскільки неузгодженість стану між компонентами або між клієнтом і сервером може порушити цілісність даних [2]. В екосистемі Angular ця проблема вирішується за допомогою реактивного програмування із використанням бібліотеки RxJS [7]. Застосування патерна Observables дозволяє декларативно моделювати асинхронні процеси, формуючи єдиний потік даних із відповідей REST API, WebSocket-з'єднань і подій від користувача [8]. Такий підхід усуває необхідність імперативного відстеження змін, оскільки дані автоматично поширюються системою через підписки. Це забезпечує узгодженість і дає змогу реалізовувати складні сценарії обробки даних із мінімальними витратами

ресурсів пристрою [7]. Наприклад, у корпоративному поштовому клієнті на Angular із RxJS обробка вхідних листів організована так, що всі події об'єднуються в один потік, що дозволяє автоматично оновлювати список листів лише за виникнення змін. Система миттєво реагує на дії користувача без дублювання логіки в окремих компонентах. Використання RxJS підвищує узгодженість стану, зменшує ймовірність помилок при оновленні та полегшує супровід складних систем.

Обробка даних, ізоляція API та використання патерна «Фасад». Переходячи до аспектів інтеграції та взаємодії з даними, слід зазначити, що інтеграція із зовнішніми підсистемами та обробка складних форматів даних вимагають строгої інженерної ізоляції. Для цього впроваджуються сервіси-адаптери, які трансформують DTO (Data Transfer Objects) з сервера у внутрішні доменні моделі клієнтського застосунку. Для стандартизації взаємодії компонентів із сервісами даних та зовнішніми API застосовується патерн «Фасад» [9]. Фасад приховує складну логіку управління станом, наприклад, реалізацію через NgRx або Akita, і надає компонентам спрощений інтерфейс для взаємодії [9]. Внаслідок цього компоненти інтерфейсу залишаються абстрагованими від обробки даних. Навантаження з обробки даних перенесено на рівень сервісів.

Масштабування клієнтських систем за допомогою впровадження мікрофронтендів вирішує проблеми тісного зв'язування, характерні для монолітної клієнтської архітектури, подібно до бекенд-монолітів. Паралельна розробка ускладнюється, збільшується час компіляції та зростає ризик каскадних помилок [6, 10]. Індустрія відповіла на ці виклики впровадженням мікрофронтендів (Micro-frontends). Технології, такі як Webpack Module Federation, дозволяють розділити єдиний корпоративний інтерфейс на незалежні застосунки [6, 10]. Ці мікрозастосунки розробляються й розгортаються окремими командами, а об'єднуються у єдиний продукт під час виконання. Це відображає перенесення класичних принципів розподілених систем, таких як незалежне розгортання та ізоляція контекстів (Bounded contexts), на рівень

клієнта [1]. Однак впровадження мікрофронтендів супроводжується і недоліками: зростає складність інтеграції окремих модулів, виникають труднощі з уніфікацією стилів і бібліотек, а також необхідність синхронізувати версії залежностей. Збільшується навантаження на інфраструктуру, можливі проблеми із продуктивністю через зростання мережевих запитів або дублювання коду. Для успішного застосування підходу мікрофронтендів необхідно впроваджувати чіткі стандарти інтеграції та активно контролювати архітектурну цілісність.

Висновки: Проведений аналіз підтверджує, що сучасний front-end несе ті самі архітектурні ризики, що й серверні системи, і вимагає застосування надійних архітектурних рішень. Angular забезпечує базу для таких рішень завдяки DI, статичній типізації та компонентній моделі; RxJS перетворює асинхронні події на керовані потоки даних; патерни Фасад і Адаптер ізолюють компоненти від деталей реалізації, а мікрофронтенди переносять принцип незалежного розгортання на клієнтський рівень. Разом ці інструменти формують архітектуру, здатну зберігати підтримуваність і стійкість у довгостроковій перспективі.

Список використаних джерел:

1. Martin R. C. The Clean Architecture. 8th Light Blog, 2012. URL: https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html (дата звернення: 08.04.2026).
2. Kazman R. et al. A Holistic View of Architecture Definition, Evolution, and Analysis. Carnegie Mellon University, Software Engineering Institute, 2023. (CMU/SEI-2023-TR-004). URL: https://www.sei.cmu.edu/documents/5720/2023_005_001_983542.pdf (дата звернення: 08.04.2026).
3. Angular Documentation. Architecture Overview & Dependency Injection. Angular.dev. URL: https://angular.dev/guide/architecture (дата звернення: 08.04.2026).

4. Microsoft. The TypeScript Handbook. TypeScript Official Documentation. URL: https://www.typescriptlang.org/docs/handbook/intro.html (дата звернення: 08.04.2026).
5. Google. Angular Overview. Angular Official Documentation, 2023. URL: https://angular.dev/overview (дата звернення: 08.04.2026).
6. Microsoft. Cloud Design Patterns. Azure Architecture Center. URL: https://learn.microsoft.com/en-us/azure/architecture/patterns/ (дата звернення: 08.04.2026).
7. ReactiveX. RxJS: Reactive Extensions Library for JS. URL: https://rxjs.dev/guide/overview (дата звернення: 08.04.2026).
8. Angular University. RxJs for Beginners: A Beginner Friendly Introduction (Functional Reactive Programming for Angular Developers). URL: https://blog.angular-university.io/functional-reactive-programming-for-angular-2-developers-rxjs-and-observables/ (дата звернення: 08.04.2026).
9. Nrwl / Nx. Nx Architecture Concepts. URL: https://nx.dev/concepts/more-concepts/applications-and-libraries (дата звернення: 08.04.2026).
10. Jackson C. Micro Frontends. MartinFowler.com, 2019. URL: https://martinfowler.com/articles/micro-frontends.html (дата звернення: 08.04.2026).

УДК 004.415.2

ГІБРИДНА АРХІТЕКТУРА АДАПТИВНОГО ІНФЕРЕНСУ ДЛЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ МЕДИЧНИХ ЗОБРАЖЕНЬ

Кудінов М. А.
kudinov@gmail.com
Черкаський державний фаховий бізнес-коледж
Марченко С. В.
м.Черкаси, Україна

Сучасні підходи до автоматизованого аналізу медичних зображень базуються на застосуванні глибоких нейронних мереж, які демонструють високі показники точності при виявленні патологічних змін у різних модальностях

діагностичних досліджень. Згідно з оглядовими дослідженнями, використання моделей глибокого навчання дозволило суттєво підвищити ефективність інтерпретації медичних зображень, проте водночас поставило нові вимоги до архітектурної організації програмних систем, у яких такі моделі експлуатуються [1].

Однією з ключових проблем практичного впровадження систем медичного аналізу зображень є необхідність досягнення компромісу між швидкодією обробки та точністю виявлення патологій. Багато сучасних моделей демонструють високі показники чутливості лише за умови значних обчислювальних витрат, що обмежує їх використання в сценаріях оперативного скринінгу або потокового аналізу зображень. У свою чергу, спрощені або оптимізовані архітектури забезпечують меншу затримку інференсу, але можуть втрачати здатність до виявлення малих або складних патологічних структур. Такі суперечності детально розглядаються у роботах, присвячених застосуванню глибокого навчання в медичному аналізі зображень, де підкреслюється залежність ефективності моделей від умов їх експлуатації та характеристик даних [2].

Додатковим фактором, що ускладнює використання однієї універсальної моделі, є неоднорідність медичних зображень, отриманих із різних діагностичних пристроїв або в різних клінічних умовах. Варіативність просторової роздільної здатності, рівня шуму, контрастності та структури патологій призводить до зниження узагальнювальної здатності моделей і потребує застосування адаптивних підходів до вибору алгоритмів обробки. У цьому контексті дослідження ефективних архітектур детекції об’єктів демонструють, що масштабовані моделі можуть бути оптимізовані під різні сценарії використання шляхом балансування точності та ресурсних витрат, що відкриває можливості для комбінування декількох режимів інференсу в межах однієї системи [3].

Окрім алгоритмічних аспектів, важливим є питання експлуатаційної надійності систем медичного аналізу. У реальних умовах використання можливе

поступове зниження якості прогнозів моделей через зміну розподілів даних, появу нових типів патологій або зміну параметрів діагностичного обладнання. Це обумовлює необхідність інтеграції механізмів моніторингу якості інференсу, збору телеметрії та повторної валідації моделей у процесі експлуатації, що розглядається в сучасних підходах до організації життєвого циклу машинного навчання [4].

Звідси, аналіз наявних підходів свідчить про доцільність переходу від використання статичних схем інференсу до адаптивних архітектур, здатних динамічно змінювати стратегію обробки залежно від складності клінічного випадку та характеристик вхідних даних. Одним із перспективних напрямів є побудова гібридних систем, у яких швидкі моделі первинного скринінгу поєднуються з більш точними моделями уточнювального аналізу. Це дозволяє забезпечити одночасно прийнятну затримку обробки та підвищену чутливість до складних патологічних ознак.

Розробити архітектуру програмної системи автоматизованого аналізу медичних зображень, що реалізує гібридний вибір моделі виявлення та забезпечує баланс між швидкістю інференсу, точністю локалізації патологій і керованістю системи в умовах реального експлуатаційного середовища.

На рис. 1 подано компонентну діаграму системи автоматизованого аналізу медичних зображень із гібридним вибором моделі виявлення патологій.

Взаємодія користувача із системою здійснюється через діагностичний інтерфейс, який забезпечує ініціацію аналізу медичних зображень та перегляд отриманих результатів. Джерело медичних зображень передає дані для обробки до компонента приймання, який виконує перевірку формату вхідних даних і передає їх до компонента попередньої обробки. На цьому етапі здійснюється нормалізація інтенсивностей, масштабування та інші операції підготовки зображення до інференсу, що дозволяє зменшити вплив неоднорідності діагностичних даних.

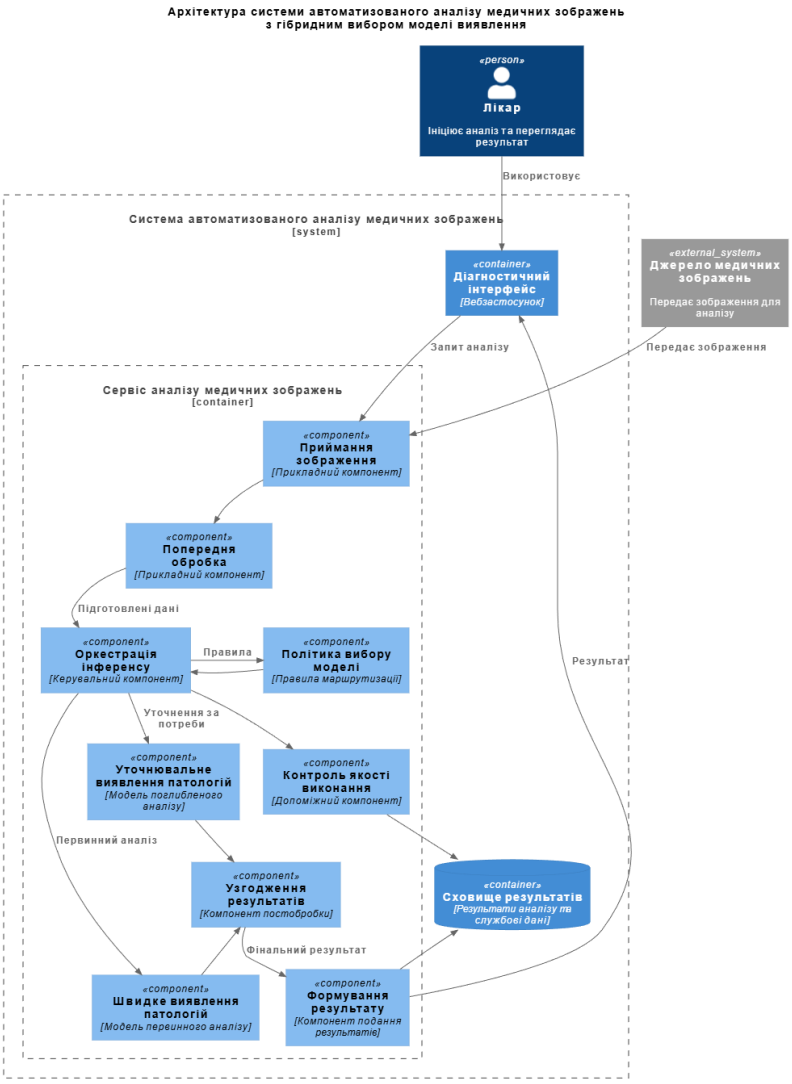


Рисунок 1 – Архітектура системи автоматизованого аналізу медичних зображень з гібридним вибором моделі виявлення

Центральним елементом запропонованої архітектури є компонент оркестрації інференсу, який координує послідовність виконання аналізу. Він взаємодіє з компонентом визначення політики вибору моделі, що формалізує критерії маршрутизації залежно від характеристик вхідних даних і результатів попереднього аналізу. На першому етапі застосовується компонент швидкого виявлення патологій, орієнтований на мінімальну затримку обробки. У разі недостатньої впевненості результатів або виявлення потенційно складних патологічних структур оркестратор ініціює запуск компонента уточнювального виявлення. Така каскадна схема дозволяє поєднати високу продуктивність первинного скринінгу з підвищеною точністю аналізу складних випадків.

Результати роботи моделей передаються до компонента узгодження результатів, який виконує їх інтеграцію, усунення конфліктних детекцій і формування узгодженого набору областей інтересу. Далі компонент формування результату забезпечує збереження отриманих даних у відповідному сховищі та передачу фінального результату користувацькому інтерфейсу.

Окрему допоміжну функцію виконує компонент контролю якості виконання, який отримує службову інформацію про перебіг інференсу та накопичує її для подальшого аналізу. Це дозволяє оцінювати ефективність роботи моделей у процесі експлуатації та своєчасно виявляти ознаки погіршення їх продуктивності.

Запропонована компонентна структура забезпечує слабке зв’язування між модулями обробки даних і прийняття рішень, що створює передумови для адаптації системи до різних клінічних сценаріїв використання та поступового розвитку її функціональності. У перспективі розвиток архітектури може передбачати інтеграцію системи з клінічними інформаційними середовищами, використання спеціалізованих сховищ метаданих і телеметрії, а також впровадження механізмів керування версіями моделей і повторної валідації. Такі розширення підвищуватимуть керованість життєвого циклу моделей і забезпечуватимуть масштабованість системи в умовах реального медичного середовища.

Список використаних джерел:

1. Litjens G., Kooi T., Bejnordi B. E., Setio A. A. A., Ciompi F., Ghafoorian M., van der Laak J. A. W. M., van Ginneken B., Sánchez C. I. A survey on deep learning in medical image analysis // *Medical Image Analysis*. 2017. Vol. 42. P. 60–88. DOI: 10.1016/j.media.2017.07.005.
2. Agneya D. A., Shekar M. S., Bharadwaj A., Vineeth N., Neelima M. L. Deep learning in medical image analysis: a survey // *Proceedings of the International Conference on Innovation and Novelty in Engineering and Technology (INNOVA)*. 2024. P. 1–5. DOI: 10.1109/INNOVA63080.2024.10847040.

3. Tan M., Pang R., Le Q. EfficientDet: scalable and efficient object detection // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 2020. P. 10781–10790. DOI: 10.1109/CVPR42600.2020.01079
4. Kreuzberger D., Kühl N., Hirschl S. Machine learning operations (MLOps): overview, definition, and architecture. *IEEE Access*. 2023. Vol. 11. P. 31866–31879. DOI:10.1109/ACCESS.2023.3262138

УДК 004.67

ПРОГРАМА ПРОГНОЗУВАННЯ ЕМОЦІЙНОГО СТАНУ ЛЮДИНИ НА ОСНОВІ АКТИВНОСТІ У СМАРТФОНІ

Воробйов Д.С.
vorobiovdmytro11@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

У сучасних умовах зростання соціально-економічних та інформаційних навантажень питання психологічного благополуччя набуває особливої актуальності. Такі явища, як тривожність, емоційне вигорання та хронічна перевтома, виходять за межі клінічної практики та стають поширеними у повсякденному житті значної частини населення.

Водночас розвиток цифрових технологій сприяє появі нових інструментів підтримки ментального здоров’я. Сучасні мобільні пристрої виконують не лише інформаційно-комунікаційну функцію, але й виступають засобом самодопомоги та психоемоційної регуляції.

Одним із прикладів сучасних цифрових рішень у сфері ментального здоров’я є застосунок FABU, розроблений компанією SUITSME (екосистема Genesis). Основною концепцією сервісу є поєднання технік саморегуляції з елементами гейміфікації.

На відміну від традиційних цифрових щоденників або трекерів настрою, FABU пропонує користувачам інноваційний підхід до вираження емоцій через

креативну діяльність, зокрема шляхом створення віртуального образу відповідно до поточного емоційного стану.

Застосунок доступний на платформах Google Play та App Store, має україномовний інтерфейс і є безкоштовним для користувачів з українськими налаштуваннями. Процес взаємодії передбачає створення емоційного профілю шляхом відповідей на запитання, що дозволяє системі визначати індивідуальні особливості користувача.

Функціональні можливості FABU включають фіксацію емоційного стану, визначення його причин, надання позитивних афірмацій, генерацію кольорових палітр для створення віртуальних образів, а також відстеження динаміки настрою за допомогою календаря. Додатково користувачам пропонуються дихальні практики та рекомендації для покращення психоемоційного стану [1].

Іншим прикладом є програма PROSIT, розроблена дослідниками університету Далхоузі (Канада), яка здійснює моніторинг ментального стану користувача на основі аналізу поведінкових даних.

Застосунок дозволяє фахівцям оцінювати стан пацієнта поза межами медичного закладу та коригувати терапевтичні заходи. PROSIT обробляє інформацію з різних джерел, зокрема дані про сон, фізичну активність, тривалість використання пристрою та комунікаційну активність.

Додатково система використовує суб’єктивні оцінки користувача, включаючи аудіозаписи та самооцінку емоційного стану, що забезпечує більш комплексний аналіз. Водночас програма розглядається як допоміжний інструмент і не може замінити професійну психотерапевтичну допомогу [2].

Серед інших популярних застосунків у сфері ментального здоров’я варто виокремити [3]:

1. MindDoc – інструмент для моніторингу емоційного стану та роботи з проявами тривожності й депресії; має статус медичного продукту.
2. Wysa – поєднує технології штучного інтелекту з психотерапевтичними підходами, забезпечуючи анонімну підтримку користувачів.

3. Spring Health Mobile – надає доступ до психологічної допомоги та інструментів подолання стресу.
4. Replika – інтерактивний цифровий співрозмовник, що адаптується до індивідуальних особливостей користувача.
5. BetterHelp – платформа онлайн-консультування, яка забезпечує доступ до професійної психологічної допомоги.

Отже, сучасні мобільні застосунки виступають ефективним інструментом підтримки ментального здоров'я, забезпечуючи можливості моніторингу емоційного стану, розвитку навичок саморегуляції та отримання психологічної допомоги. Водночас такі технології не можуть повністю замінити професійну психотерапевтичну практику, а функціонують як її доповнення, сприяючи підвищенню доступності та оперативності надання допомоги.

Список використаних джерел:

1. Застосунки для підтримки ментального здоров'я. URL: https://surl.li/bfuach (дата звернення: 23.03.2026).
2. Програма PROSIT для моніторингу психічного стану. URL: https://surli.cc/gvanrv (дата звернення: 23.03.2026).
3. Огляд застосунків для ментального здоров'я. URL: https://surl.li/zjzkex (дата звернення: 23.03.2026).

УДК 004.8:004.415

ІНТЕГРАЦІЯ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ В ПРОЦЕСИ АВТОМАТИЗАЦІЇ ІНЖЕНЕРІЇ ВИМОГ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Поліщук О. В.
olenapolishchuk022@gmail.com
Черкаський державний фаховий бізнес-коледж
Марченко С. В.
м.Черкаси, Україна

Створення сучасного програмного забезпечення – це багаторівневий процес, де навіть незначна неточність на етапі проєктування може призвести до критичних наслідків. Інженерна практика доводить: етап збору вимог (Requirements Engineering) слугує базовим фундаментом розробки [3]. Практичний досвід IT-компаній підтверджує, що виправлення хибно зрозумілої вимоги під час написання коду чи тестування потребує найбільших фінансових та часових витрат [7]. Тому життєздатність проєкту прямо залежить від якості складеної специфікації.

Традиційно технічні завдання (Software Requirements Specifications) формулюються природною мовою. Це логічний компроміс, який дозволяє замовнику та команді знайти спільне розуміння задачі [6]. Водночас такий формат створює значну проблему. Природна мова відрізняється високим рівнем неоднозначності, містить надмірну інформацію, через що критичні технічні нюанси можуть просто загубитися [1]. Аби програмісти отримали чітку структуру, системні аналітики змушені витрачати години на ручний переклад цих об’ємних текстів у візуальні моделі (наприклад, UML-діаграми). Такий рутинний процес є тривалим і часто супроводжується помилками через «людський фактор» [2]. З огляду на це, автоматизація трансформації сирого тексту в графічні моделі стає необхідним кроком. Це не лише прискорить темпи розробки, але й суттєво зменшить імовірність архітектурних прорахунків.

Проаналізувати наявні інженерні методи обробки вимог, оцінити можливості штучного інтелекту в задачах аналізу тексту та спроєктувати

архітектуру системи для автоматичного перетворення технічних завдань у код побудови діаграм.

Останніми роками підходи до проєктування зазнали суттєвих змін. Індустрія поступово відмовляється від ручного малювання схем, переходячи до парадигми «Діаграми як код» (Diagrams as Code). Концепція полягає в тому, що інженер пише скрипт, а система генерує візуальну схему. Інструменти на зразок PlantUML або Mermaid.js набули широкого поширення, оскільки дозволяють зберігати історію змін архітектури безпосередньо в репозиторіях Git разом із первинним кодом.

Попри всі переваги, ці інструменти мають серйозне обмеження: вони працюють виключно як візуалізатори. Їм потрібен ідеальний машинний синтаксис, і вони не здатні самостійно інтерпретувати звичайний текст. Відповідно, найскладніша аналітична частина – прочитати вимоги, виділити бізнес-логіку та написати код діаграми – все ще залишається зоною відповідальності людини.

Вирішення цієї проблеми стало можливим завдяки великим мовним моделям (LLM) [9]. У їхній основі лежить архітектура Transformer, яка використовує механізм «уваги» (Self-Attention). На відміну від класичних алгоритмів, трансформери аналізують весь документ цілісно. Вони здатні знаходити взаємозв’язки між вимогами, навіть якщо ті знаходяться в різних розділах специфікації [4].

На практиці процес виглядає так: після передачі абзацу з бізнес-вимогами на вхід моделі (наприклад, сімейства GPT), алгоритм самостійно ідентифікує акторів системи, бази даних та правила їхньої взаємодії [1]. Після цього миттєво генерується синтаксис, який відразу трансформується у графічну схему [2].

Проте застосування базових моделей має свої ризики. Прямий запит до нейромережі часто призводить до генерації синтаксичних помилок або додавання зайвих коментарів (так званих «галюцинацій») [9]. Для забезпечення детермінованого результату фахівці застосовують Prompt Engineering (інженерію підказок) [5]. Нейромережі надають кілька еталонних

прикладів очікуваного результату (Few-Shot Prompting) [8] та встановлюють суворі системні обмеження. Це змушує алгоритм ігнорувати розмовний стиль і генерувати виключно чистий, готовий до компіляції код діаграми [5].

Для практичного застосування проєктується мікросервісний вебдодаток. Серверна частина (backend) відповідає за взаємодію з ШІ через API: вона отримує згенерований код, очищає його від можливих артефактів і передає на клієнтську сторону (frontend). Браузер виконує лише рендеринг графіки. Завдяки цьому системний аналітик уникає необхідності працювати з промптами чи синтаксисом − він просто вводить текст і отримує готову архітектурну модель.

Точність генерації можна додатково підвищити шляхом донавчання (fine-tuning) моделі на базах реальної технічної документації [10]. Завдяки цьому штучний інтелект краще адаптується до специфічної інженерної термінології, а відсоток синтаксичного браку зводиться до мінімуму [1].

Впровадження великих мовних моделей у процес інженерії вимог дозволяє суттєво оптимізувати проєктування програмного забезпечення. Штучний інтелект бере на себе роль інтелектуального транслятора між природною мовою замовника та строгим кодом візуальних моделей. Це не лише економить ресурси команди, але й допомагає виявляти логічні прогалини в архітектурі ще до початку написання коду. Зрештою, це створює надійне підґрунтя для переходу до повної кодогенерації додатків на основі затверджених текстових специфікацій.

Список використаних джерел:

1. Ray T., Cole A., Pinon Fischer B. F., White A. P., Mavris D. N. Agile Methodology for the Standardization of Engineering Requirements Using Large Language Models. *Systems*. 2023. Vol. 11, No. 7. Art. no. 352. URL: https://doi.org/10.3390/systems11070352 (дата звернення: 16.03.2026).
2. Generative AI in Software Requirements Engineering: Opportunities, Challenges, and Future Directions / G. Boussaidi et al. *HAL open science*. 2024.

URL: https://u-bourgogne.hal.science/LABSOC/hal-04483279v1 (дата звернення: 16.03.2026).

3. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. 4th ed. / ed. by D. D. Walden et al. Hoboken, NJ : John Wiley & Sons, 2015. 304 p. URL: https://doi.org/10.1002/9781119287552 (дата звернення: 16.03.2026).
4. Vaswani A., Shazeer N., Parmar N., et al. Attention Is All You Need. *Advances in Neural Information Processing Systems (NeurIPS)*. 2017. Vol. 30. URL: https://arxiv.org/abs/1706.03762 (дата звернення: 16.03.2026).
5. White J., Fu Q., Hays S., et al. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *arXiv preprint*. 2023. URL: https://arxiv.org/abs/2302.11382 (дата звернення: 16.03.2026).
6. ISO/IEC/IEEE 29148:2018. Systems and software engineering – Life cycle processes – Requirements engineering. IEEE, 2018. 104 p. URL: https://doi.org/10.1109/IEEESTD.2018.8559686 (дата звернення: 16.03.2026).
7. Pohl K. Requirements Engineering: Fundamentals, Principles, and Techniques. Berlin: Springer Publishing, 2010. 814 p. URL: https://doi.org/10.1007/978-3-642-12578-2 (дата звернення: 16.03.2026).
8. Language Models are Few-Shot Learners / T. Brown et al. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 1877–1901. URL: https://arxiv.org/abs/2005.14165 (дата звернення: 16.03.2026).
9. Sparks of Artificial General Intelligence: Early experiments with GPT-4 / S. Bubeck et al. *arXiv preprint*. 2023. URL: https://arxiv.org/abs/2303.12712 (дата звернення: 16.03.2026).
10. Ouyang L., Wu J., Jiang X., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*. 2022. Vol. 35. P. 27730–27744. URL: https://arxiv.org/abs/2203.02155 (дата звернення: 16.03.2026).

УДК 004.056.52:004.77

СИСТЕМА МОНІТОРИНГУ КОМПРОМЕТАЦІЇ ОБЛІКОВИХ ДАНИХ

Гетьман І.І.
getman_II@gmail.com
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м.Черкаси, Україна

Актуальність проблеми витоку облікових даних у відкритих джерелах зумовлена постійним збільшенням кількості інтернет-сервісів, накопиченням персональної інформації, поширенням кіберзлочинності та використанням компрометованих даних у шахрайських схемах. Витік облікових даних призводить до фінансових втрат користувачів, компрометації корпоративних систем, втрати конфіденційності та зниження довіри до цифрових сервісів [1]. Аналіз сучасних методів моніторингу дозволяє класифікувати джерела витоків на публічні бази даних, форуми, сайти з витоками, а також платні та безкоштовні REST API сервіси перевірки компрометацій [2].

Функціональна архітектура системи, що розробляється, базується на високоефективній мові програмування Python, що забезпечує стабільність обчислювальних процесів та сучасну підтримку асинхронності. Взаємодія з користувачем реалізована через інтерактивний інтерфейс Telegram-бота, побудований на базі фреймворку Aiogram, який дозволяє гнучко опрацьовувати події в режимі реального часу. Для забезпечення цілісності обміну даними із зовнішніми ресурсами інтегровано бібліотеку requests, яка виконує роль сполучного механізму з різноманітними REST API сервісами.

Програмна структура системи спроєктована за модульним принципом, що дозволяє досягти високого рівня автономності окремих елементів та спрощує подальше масштабування продукту. Зокрема, архітектура передбачає чіткий поділ на рівні логіки, де виокремлюються блоки обробки вхідних повідомлень, модулі формування запитів до зовнішніх серверів та компоненти валідації отриманих результатів. Такий підхід не лише підвищує відмовостійкість системи за рахунок ізоляції можливих помилок у межах окремих модулів,а й дозволяє

інтегрувати нову функціональність шляхом розширення існуючих модулів без ризику порушення стабільної роботи фундаментальних алгоритмів. Система складається з наступних модулів:

1. Модуль збору даних – забезпечує регулярне сканування відкритих джерел, включаючи форуми, сайти з витоками та API сервісів; виконує парсинг та структурування отриманої інформації; застосовує фільтри для виділення релевантних даних.
2. Модуль обробки даних – перевіряє коректність отриманих даних, визначає тип витоку, ступінь ризику та пріоритет оповіщення; реалізує алгоритми валідації та нормалізації даних, включаючи перевірку формату логінів та хешів паролів.
3. Модуль логування та аудиту – фіксує всі події системи, веде історію перевірок, зберігає статистику ефективності та точності моніторингу; підтримує інтеграцію з Telegram-ботом для оперативного інформування.
4. Telegram-бот – надає користувачу можливість запускати перевірки, отримувати результати, переглядати історію моніторингу, а також конфігурувати частоту перевірок та типи джерел.
5. Модуль тестування – реалізує сценарії з використанням Mock-об’єктів для безпечного моделювання витоків, емулює роботу API та бази даних без доступу до реальних облікових даних.

Система підтримує інтеграцію з платними REST API сервісами, використовуючи захищене підключення через HTTPS та токени доступу. Підключення здійснюється в тестовому режимі перед дипломним захистом, що дозволяє перевірити точність та ефективність сервісів, порівняти локальне тестування та реальні дані [3].

Для забезпечення високої точності оцінювання результатів моніторингу в системі впроваджено спеціалізовані алгоритми автоматизованої класифікації. Ці інструменти дозволяють диференціювати виявлені витоки за критеріями потенційного ризику та встановлювати пріоритетність подальшого оповіщення користувачів. Процес обробки інформації базується на комплексному аналізі

критичності скомпрометованих даних, що дозволяє формувати чітку ієрархію загроз.

Усі отримані відомості підлягають суворій структуризації: фіксується точна дата виявлення інциденту, ідентифікується тип джерела та обчислюється обсяг скомпрометованих облікових записів. Підсумковим етапом аналізу є генерація адаптивних рекомендацій, спрямованих на нейтралізацію безпекових ризиків для кінцевого споживача. Деталізований приклад результатів такої аналітичної обробки подано в табл. 1.

Таблиця 1 – Результати верифікації та класифікації виявлених інцидентів інформаційної безпеки

№	Джерело	Кількість записів	Тип даних	Ступінь ризику
1	ForumXYZ	25	Email + Password	Високий
2	Pastebin	12	Email	Середній
3	API Service	40	Email + Hash	Високий

Тестування системи включає модульне та інтеграційне тестування. Модульне тестування перевіряє роботу кожного компонента окремо, а інтеграційне – взаємодію модулів між собою та з Telegram-ботом. Використання Mock-об’єктів дозволяє створювати контрольовані сценарії витоків для оцінки реакції системи без ризику для реальних даних [4].

Результати показали, що застосування модульної архітектури та інтеграція з Telegram-інтерфейсом дозволяє ефективно моніторити витоки облікових даних, надавати швидке сповіщення користувачам та вести історію перевірок для подальшого аналізу. Перспективи розвитку включають розширення бази джерел, застосування алгоритмів машинного навчання для класифікації ризиків та автоматичного визначення типу витоку, а також інтеграцію з корпоративними системами безпеки [5].

Список використаних джерел

1. Aiogram Documentation. URL: https://docs.aiogram.dev (дата звернення 03.03.2026)
2. Have I Been Pwned API Documentation. URL: https://haveibeenpwned.com/API/v3 (дата звернення 05.03.2026)
3. Mock Library Documentation. URL: https://docs.python.org/3/library/unittest.mock.html (дата звернення 08.03.2026)
4. OWASP. Top 10 Security Risks for Web Applications. URL: https://owasp.org (дата звернення 10.03.2026)
5. Python 3.11 Documentation. URL: https://www.python.org/doc/ (дата звернення 12.03.2026)

УДК 004.41

ОСОБЛИВОСТІ ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЇ ВЕБ-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ ШАБЛОНУ MVT

Синьогуб А. Р.
alina.sinegu@icloud.com
Черкаський державний бізнес-коледж
Подорошко Д. І.
м. Черкаси, Україна

У роботі розглянуто особливості проєктування та реалізації веб-орієнтованої інформаційної системи «Beauty Time» на основі шаблону MVT із використанням фреймворку Django. Актуальність теми зумовлена потребою створення масштабованих, безпечних і зручних веб-застосунків у сфері послуг, де важливо забезпечити надійну обробку записів клієнтів, мінімізувати помилки під час бронювання та підвищити якість обслуговування.

Проєктується веб-орієнтована інформаційна система для салону краси, яка призначена для автоматизації процесу онлайн-запису клієнтів на послуги. Розробка такої системи обумовлена необхідністю переходу від неефективного ручного менеджменту записів до єдиного цифрового середовища, що дозволяє

уникнути накладання бронювань, оптимізувати роботу адміністратора та підвищити зручність взаємодії користувача із сервісом.

З метою підвищення ефективності розробки та перевірки доцільності створення системи було обрано підхід MVP. Це обґрунтовується тим, що основне призначення MVP полягає у тестуванні гіпотез та перевірці затребуваності задуманого продукту на ринку [1], що дозволяє оцінити практичну цінність системи ще на ранніх етапах її створення та зменшити ризики надмірних витрат ресурсів.

Разом із цим було розглянуто альтернативні підходи до розробки веб-застосунків, зокрема використання монолітних архітектур без чіткого розділення компонентів або клієнт-орієнтованих рішень із надмірним зміщенням логіки на фронтенд. Проте такі підходи є менш доцільними для систем онлайн-запису, оскільки ускладнюють підтримку, тестування та забезпечення цілісності даних. Це підтверджується тим, що хоча MVC спочатку був розроблений для настільних обчислень, він широко використовувався як дизайн для застосунків Всесвітньої павутини основними мовами програмування. Було створено кілька веб-фреймворків, які забезпечують виконання цього шаблону [2]. Отже, використання архітектурних підходів із чітким розмежуванням відповідальності між компонентами є доцільним і для сучасних веб-орієнтованих інформаційних систем.

Крім того, відсутність чіткого розмежування рівнів логіки, даних і представлення може призводити до зростання кількості помилок та ускладнення масштабування системи.

Метою дослідження є проєктування та реалізація веб-орієнтованої інформаційної системи онлайн-запису на послуги салону краси на основі шаблону MVT із використанням фреймворку Django.

Для досягнення поставленої мети було визначено такі завдання:

– проаналізувати особливості застосування шаблону MVT у веб-розробці;

- дослідити доцільність використання Django для побудови системи онлайн-запису;
- спроєктувати структуру даних системи;
- реалізувати серверну логіку обробки записів клієнтів;
- створити адаптивний користувацький інтерфейс;
- забезпечити перевірку доступності часових слотів і запобігання конфліктам;
- протестувати основні сценарії роботи системи.

Вибір фреймворку Django та шаблону MVT у межах даного дослідження є обґрунтованим з урахуванням специфіки задачі. Для систем онлайн-запису критично важливими є надійна обробка даних, швидкість розробки та можливість масштабування. Це узгоджується з однією з фундаментальних засад Django – loose coupling and tight cohesion [3], тобто слабкою зв'язаністю компонентів і високою внутрішньою узгодженістю системи. Крім того, у документації Django підкреслюється, що фреймворк має забезпечувати incredibly quick web development [3], що є особливо важливим при реалізації MVP веб-орієнтованої інформаційної системи. Для систем онлайн-запису критично важливими є надійна обробка даних, швидкість розробки та можливість масштабування. На основі аналізу існуючих підходів до створення подібних систем було встановлено, що використання Django дозволяє ефективно реалізувати необхідний функціонал із мінімальними витратами часу на розробку базових компонентів. Це підтверджується тим, що використання фреймворків, таких як Django, дозволяє швидко розробляти безпечні й надійні веб-застосунки стандартизованим способом, без необхідності повторно винаходити колесо.

Таким чином, застосування даних методів у цьому проєкті є доцільним рішенням, що забезпечує структурованість коду, зручність підтримки та можливість подальшого розширення функціоналу системи.

Крім того, використання підходу MVP дозволяє суттєво скоротити час виходу продукту на ринок та отримати зворотний зв'язок від користувачів на ранніх етапах. Зокрема, швидкий запуск – MVP дозволяє вийти на ринок за

кілька тижнів або місяців замість років розробки, що є важливим фактором при створенні веб-сервісів у сфері послуг.

У процесі реалізації інформаційної системи «Beauty Time» було спроєктовано ключові сутності предметної області: користувач, послуга, запис і повідомлення з форми зворотного зв'язку. Рівень даних (Model) реалізовано засобами Django ORM, що забезпечує цілісність даних і контроль обмежень. Рівень логіки (View) відповідає за обробку HTTP-запитів, валідацію даних, створення, редагування та скасування записів, а також перевірку доступності часових слотів. Рівень представлення (Template) забезпечує формування динамічних веб-сторінок відповідно до дій користувача.

У розробленій системі реалізовано такі функціональні можливості: реєстрація та авторизація користувачів; перегляд переліку послуг; онлайн-запис на обрану послугу; автоматична перевірка зайнятості часових слотів; відображення доступного часу; особистий кабінет користувача для керування записами; форма зворотного зв'язку; адміністративна панель для керування даними системи.

Особливу увагу було приділено механізму запобігання конфліктам у розкладі. У системі реалізовано серверну перевірку доступності обраного часового слоту та алгоритм підбору вільного часу, що дозволяє уникнути накладання записів і підвищує надійність функціонування системи.

У результаті виконаної роботи було розроблено повнофункціональну веб-орієнтовану інформаційну систему «Beauty Time». Під час тестування було перевірено основні сценарії роботи системи, включаючи реєстрацію користувачів, створення та керування записами, обробку помилкових ситуацій і роботу адміністративної панелі. Отримані результати підтвердили коректність реалізації ключових модулів, стабільність роботи системи та ефективність обраного підходу.

Таким чином, використання шаблону MVT у поєднанні з фреймворком Django є доцільним і обґрунтованим для розробки веб-орієнтованих інформаційних систем у сфері послуг. Такий формат організації системи є

невід’ємною складовою сучасних веб-застосунків, оскільки забезпечує надійність, гнучкість, зручність подальшого розвитку та відповідність вимогам реальних практичних задач.

Список використаних джерел

1. Модель–вигляд–контролер – ВікіпедіяDevZone. Посібник по Django для початківців. Частина 1. URL: https://devzone.org.ua/post/posibnyk-po-django-dlia-pochatkivtsiv-chastyna-1 (дата звернення: 29.03.2026).
2. Wikipedia. Model–view–controller . URL: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller (дата звернення: 30.03.2026).
3. Django documentation. Design philosophies. URL: https://docs.djangoproject.com/en/dev/misc/design-philosophies/ (дата звернення: 17.03.2026).
4. Wezom. Фреймворк Python Django для створення сайтів . URL: https://wezom.com.ua/ua/blog/razrabotka-sajtov-na-python-django (дата звернення: 30.03.2026).
5. Gan P., Gu Z., Zou H., Zhu T., Li Z. A Django-Based Modeling Platform for Predicting Soil Moisture in Agricultural Fields // Water. 2025. Vol. 17, No. 12. Art. 1753. URL: https://www.mdpi.com/2073-4441/17/12/1753 (дата звернення: 30.03.2026)

УДК 004.415.2

АРХІТЕКТУРА ВЕБПЛАТФОРМИ ПРОВЕДЕННЯ ОНЛАЙН-ВІКТОРИН У РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Маренич Ф. А.
0668090426fed@gmail.com
Черкаський державний фаховий бізнес-коледж
Марченко С. В.
м.Черкаси, Україна

Інтерактивні цифрові платформи дедалі ширше застосовуються в освітньому середовищі для підтримки активних форм навчання та підвищення залученості учасників. Одним із найбільш поширених форматів є онлайн-вікторини, що поєднують перевірку знань із елементами гейміфікації. У науковій літературі гейміфікація визначається як використання елементів ігрового дизайну в неігрових контекстах, а її практична цінність полягає у стимулюванні мотивації та підтримці активної взаємодії користувачів із цифровим середовищем [1]. Водночас для платформ цього класу вирішальними є не лише мотиваційні механіки, а й архітектурні властивості системи, передусім здатність підтримувати узгоджений стан гри між усіма учасниками.

Ключовою технологічною вимогою до платформ проведення вікторин є взаємодія у режимі реального часу, оскільки під час гри активне запитання, відповіді учасників і таблиця лідерів мають синхронно оновлюватися для всіх клієнтів. Для таких сценаріїв традиційна модель періодичних HTTP-запитів є менш ефективною через надлишкові мережеві накладні витрати, тоді як WebSocket-підхід забезпечує сталі двонаправлені з’єднання і меншу затримку передавання подій [2]. Аналіз наявних SaaS-платформ для онлайн-вікторин свідчить, що їхні функціональні можливості часто супроводжуються обмеженнями щодо конфігурації, масштабування та керування логікою сесії. Це зумовлює потребу в архітектурному рішенні, оптимізованому саме для інтенсивного обміну короткими подіями в межах керованої ігрової сесії.

Розробити архітектуру вебплатформи проведення онлайн-вікторин у режимі реального часу, що забезпечує синхронізацію стану гри між учасниками та підтримує масштабування кількості підключених клієнтів.

Архітектуру запропонованої платформи представлено за допомогою C4-моделі, яка використовується для ієрархічного опису програмних систем і дає змогу чітко відобразити взаємодію користувачів, клієнтського застосунку та серверних компонентів (рис. 1). Користувачі системи поділяються на дві ролі: організатор, який створює вікторини та керує ігровою сесією, і гравці, які приєднуються до гри за PIN-кодом. Клієнтська частина взаємодіє з сервером двома каналами: REST API використовується для роботи зі статичними даними, тоді як передавання ігрових подій реалізовано через SignalR.

Серверна частина платформи реалізована як модульний моноліт. Такий вибір є обґрунтованим саме для системи, у якій основне навантаження формується не великою кількістю незалежних бізнес-процесів, а швидким циклом обробки коротких подій у межах однієї сесії. На відміну від мікросервісної архітектури, модульний моноліт у такому випадку дає змогу уникнути міжсервісних мережевих викликів, зменшити інфраструктурну складність і зберегти чітке логічне розмежування відповідальностей усередині застосунку. Сучасні дослідження також фіксують зворотний рух частини систем від мікросервісів до монолітів або модульних монолітів у випадках, коли накладні витрати на розподіленість перевищують виграш від неї [4].

До основних компонентів серверної частини належать REST API Controllers, SignalR Quiz Hub, Game State Manager і Data Access Layer. У цій конфігурації критично важливим є те, що архітектура фактично реалізує server-authoritative підхід: єдиним джерелом істини щодо стану гри виступає сервер. Усі події, ініційовані клієнтами, спочатку валідуються та обробляються на сервері, і лише після цього оновлений стан транслюється учасникам. Для сценарію онлайн-вікторини це принципово, оскільки усуває розбіжності між клієнтами, спрощує контроль таймерів, порядку переходів між запитаннями та алгоритмів нарахування балів.

Центральним компонентом системи є Game State Manager, який відповідає за збереження транзиторного стану активних ігор: поточного запитання, балів гравців, таймерів, службових прапорців сесії. Критичний аналіз цього рішення показує, що винесення такого стану безпосередньо в постійне сховище на кожній події створило б надлишкові I/O-затримки і погіршило б часові характеристики системи. Тому використання in-memory моделі для активних сесій є архітектурно виправданим: база даних у такому разі зберігає лише персистентні сутності та результати завершених ігор, тоді як оперативний ігровий цикл не залежить від швидкості дискових операцій. Це рішення не усуває потреби в подальшому масштабуванні, однак на рівні прототипу і цільового сценарію воно забезпечує кращий баланс між продуктивністю, простотою реалізації та керованістю системи [3; 4].

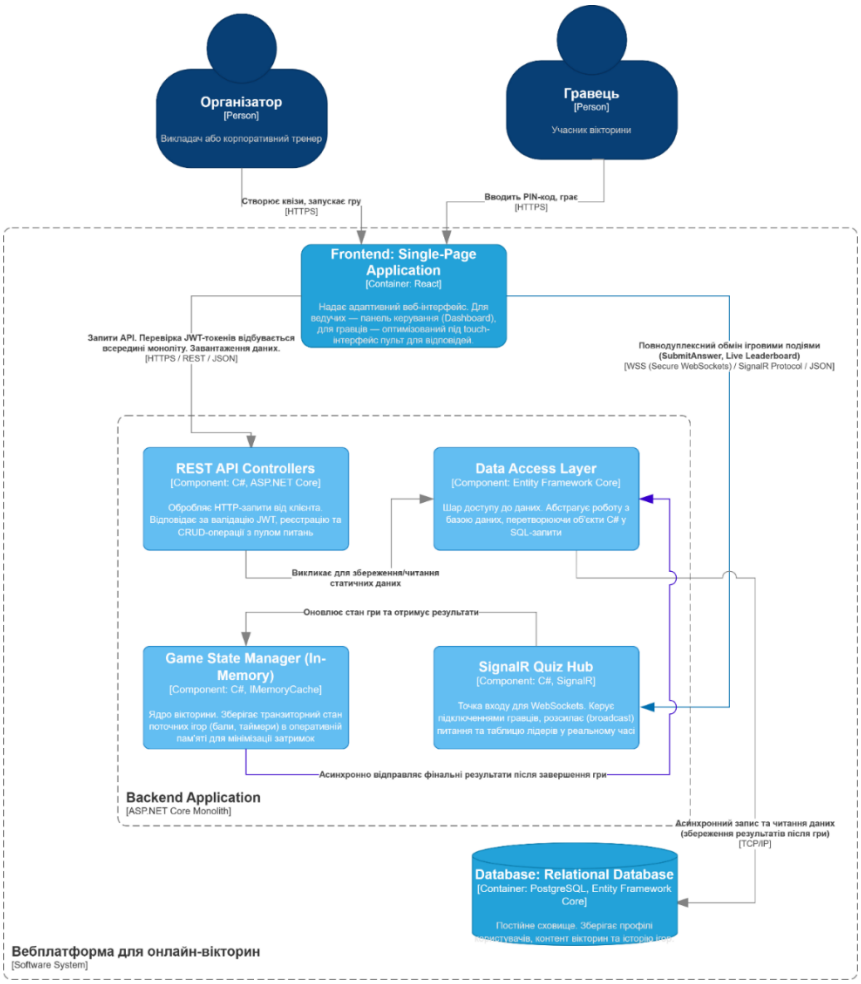


Рисунок 1 – Компонентна архітектура вебплатформи за нотацією C4

Передавання ігрових подій між сервером і клієнтами здійснюється через SignalR, який надає прикладний рівень для організації real-time взаємодії, підтримує хаби, групування підключень і механізми повторного з’єднання [5]. У межах запропонованої архітектури це дає змогу ізолювати події окремих ігрових сесій і транслювати їх лише відповідним учасникам. Таким чином, C4-діаграма не лише описує складники системи, а й відображає її головну архітектурну ідею: розмежування статичного API, realtime-каналу та окремого шару керування станом гри.

У роботі запропоновано архітектуру вебплатформи проведення онлайн-вікторин у режимі реального часу. Її побудовано як modular monolith із чітким відокремленням REST-взаємодії, каналу передавання подій у режимі реального часу та модуля керування станом гри. Критичний аналіз показав, що для системи такого класу обраний підхід є обґрунтованішим за мікросервісну декомпозицію на ранньому етапі, оскільки дозволяє зменшити латентність, уникнути надмірної інфраструктурної складності та забезпечити узгоджений стан ігрової сесії.

Список використаних джерел:

1. Deterding S., Dixon D., Khaled R., Nacke L. From game design elements to gamefulness: defining gamification // Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. Tampere, 2011. P. 9-15. DOI: 10.1145/2181037.2181040.
2. Pimentel V., Nickerson B. G. Communicating and displaying real-time data with WebSocket // IEEE Internet Computing. 2012. Vol. 16, No. 4. P. 45-53. DOI: 10.1109/MIC.2012.64.
3. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th ed. Boston : Addison-Wesley Professional, 2021. 464 p.
4. Su R., Li X., Taibi D. Back to the future: from microservice to monolith // Electronics. 2024. Vol. 13, No. 8. Art. 1452. DOI: 10.3390/electronics13081452.

5. Microsoft. Overview of ASP.NET Core SignalR. URL: https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-10.0 (дата звернення: 07.03.2026).

УДК 004.932:681.2

ІНЖЕНЕРІЯ ПОРТАТИВНИХ ЗАСОБІВ ВИМІРЮВАННЯ ТА ОПРАЦЮВАННЯ ДАНИХ

Євтушенко Д. В.
yevtushenkodv@outlook.com
Черкаський державний фаховий бізнес коледж
Фальченко Н. Г.
м.Черкаси, Україна

У сучасній комп’ютерній інженерії все більшої популярності набувають автономні пристрої моніторингу з можливістю обробки даних у реальному часі. Однією з ключових складових таких систем є модуль збору даних, який не лише відповідає за взаємодію з сенсорами, а й виконує функції з фільтрації сигналів, керування енергоспоживанням та забезпечення обміну інформацією з зовнішніми мережами. У портативній системі збору даних апаратно-програмна частина відіграє критичну роль у створенні точного, енергоефективного та надійного інструменту вимірювання [1].

Основні завдання розробки включають: обґрунтування компонентної бази, розробку структурної схеми, реалізацію алгоритмів первинної обробки сигналів, організацію передачі даних, а також забезпечення високої автономності пристрою [2].

Система реалізована на базі енергоефективного мікроконтролера, а взаємодія з сенсорним обладнанням базується на використанні цифрових інтерфейсів I2C та SPI. Головною перевагою такого підходу є швидкість зчитування та можливість прямого отримання даних без складних аналогових перетворень. Усі операції з опитування датчиків та локальних обчислень виконуються з високою частотою дискретизації. Передача обробленої

інформації до зовнішнього шлюзу здійснюється через бездротові протоколи, де дані структуруються для подальшого аналізу [3].

Особлива увага приділялася реалізації алгоритмів цифрової фільтрації та методів компенсації похибок сенсорів, що дозволяє уникнути шумів і випадкових сплесків при візуалізації. Це критично важливо в системах прецизійного моніторингу, де точність вимірювання є визначальною. Крім того, було реалізовано систему керування живленням шляхом використання режимів глибокого сну мікроконтролера між циклами вимірювань [4].

Функціонал системи включає наступні елементи: модуль зчитування фізичних величин, блок контролю напруги живлення, таймер сесій моніторингу, систему сповіщень про критичні стани, а також інтерфейс локального відображення даних. Усі вузли спроєктовано згідно з принципами ергономіки та компактності для забезпечення зручності використання у портативному форматі [5].

Логіка обробки сигналів, керування перериваннями та процеси ініціалізації периферії реалізовано через вбудоване програмне забезпечення (Firmware), із застосуванням методів буферизації даних для уникнення втрат під час передачі. Для звукового або візуального інформування про робочі стани додано систему індикації, яка активується у відповідь на події системи.

Додатково реалізовано систему логування параметрів роботи пристрою з метою аналізу стабільності та енергоаудиту. Це включає реєстрацію часу активної роботи, споживання струму в різних режимах та діагностику помилок зв’язку, що дає змогу покращити надійність системи в майбутніх релізах.

Тестування прототипу проводилося в різних умовах експлуатації та на джерелах живлення з різною ємністю. Результати підтвердили стабільну роботу системи навіть при низькому рівні заряду акумулятора. Під час сесій тривалого моніторингу пристрій демонстрував стабільний цикл опитування сенсорів із високою точністю отриманих даних.

У перспективі планується реалізація хмарного сервісу для віддаленого зберігання великих масивів даних, підтримка додаткових протоколів зв’язку,

можливість оновлення ПЗ через повітря, а також додавання системи локального збереження результатів на зовнішні носії пам'яті.

Таким чином, розроблена портативна система збору та обробки даних сенсорів є сучасним прикладом вбудованого рішення, що поєднує автономність, гнучкість і точність. Реалізовані технічні рішення дозволяють забезпечити якісний моніторинг незалежно від умов, у яких працює пристрій.

Список використаних джерел:

1. Андрієвський Б. М. Упровадження сучасних систем моніторингу в освіту: проблеми та перспективи. Інформаційні технології в освіті. 2013. Вип. 14. С. 7–10.
2. Панченко Т. та ін. Огляд портативних вимірювальних систем та їх постачальників. InterConf. 2024. № 43(193). С. 550–559.
3. Бунке О. С. Ефективні сценарії використання хмарних технологій на підприємстві. Вчені записки ТНУ імені В. І. Вернадського. Серія: технічні науки. 2020. Т. 31. № 6. С. 44–49.
4. Аналіз поняття портативних технологій: види та категорії / О. Андрощук та ін. Молодий вчений. 2021. № 6 (94). С. 83–87.
5. Ястремська О. М., Стадниченко А. В., Колобов І. Ю. Аналіз впливу технологій обчислень на стратегічне управління конкурентоспроможністю підприємств. Академічні візії. 2024. Вип. 27.

УДК 004.415.2:004.8

ІНЖЕНЕРНІ ПАТЕРНИ ПРОЄКТУВАННЯ ТА ВІЗУАЛІЗАЦІЙНІ ПІДХОДИ В СИСТЕМАХ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ

Кроть В.С.
vktr244@gmail.com
НТУУ «Київський політехнічний інститут
ім. Ігоря Сікорського»
Марченко С. В.
Київ, Україна

Генеративний штучний інтелект стрімко інтегрується у прикладні програмні продукти: чат-боти, інструменти генерації коду та агентні системи, здатні виконувати складні багатокрокові задачі. Разом із тим розробники систематично стикаються з повторюваними інженерними проблемами – нестабільністю відповідей моделей, високими витратами обчислювальних ресурсів, галюцинаціями, складністю контролю поведінки та забезпеченням безпечності результатів генерації.

В експлуатаційній практиці такі проблеми проявляються як труднощі масштабування GenAI-рішень, обмежена відтворюваність результатів і потреба у додаткових механізмах моніторингу та аналізу процесів генерації. Повторюваний характер цих викликів свідчить про формування нового класу архітектурних рішень – патернів проєктування GenAI-систем, які забезпечують інтеграцію недетермінованих когнітивних сервісів у традиційні програмні архітектури.

Проаналізувати інженерні патерни проєктування систем генеративного штучного інтелекту, визначити їхню роль у підвищенні надійності та ефективності програмних продуктів, а також дослідити можливості використання візуалізаційних підходів для контролю та оптимізації процесів генерації.

Патерни GenAI-систем формують інженерний шар взаємодії з недетермінованими моделями машинного навчання, що зміщує фокус проєктування з організації внутрішньої структури коду на управління

контекстом виконання, трасування процесів генерації та оптимізацію ресурсних характеристик системи.

Промпт-орієнтовані підходи, зокрема Chain-of-Thought, Few-Shot Prompting і Role Prompting, спрямовані на структуризацію запиту до мовної моделі з метою підвищення точності відповідей [1]. Вони фактично виконують функцію декларативного програмування поведінки моделі. Водночас використання складних промпт-патернів призводить до збільшення довжини контексту, що негативно впливає на затримки виконання та вартість обробки запитів у виробничих середовищах. Таким чином виникає інженерний компроміс між стабільністю результату та ефективністю використання обчислювальних ресурсів.

Ефективність GenAI-систем значною мірою визначається управлінням контекстним вікном моделі. Практичними рішеннями є резюмування попередніх повідомлень, кешування системних інструкцій та декомпозиція задач на незалежні підзапити. У масштабних системах такі механізми реалізуються у вигляді окремого програмного шару керування контекстом, що дозволяє балансувати між якістю генерації та ресурсною ефективністю.

Патерн роботи зі знаннями Retrieval-Augmented Generation забезпечує підвантаження релевантних документів із зовнішніх сховищ у контекст моделі, що дозволяє зменшити ризик галюцинацій і підвищити актуальність відповідей [2]. Разом із тим RAG-архітектури породжують нові виклики, пов’язані з вибором стратегій пошуку, синхронізацією джерел знань та необхідністю візуального аналізу залежностей між retrieved-контекстом і результатами генерації.

Оскільки мовні моделі не мають довготривалої пам’яті, архітектори GenAI-систем реалізують її програмно. Короткотривала пам’ять (STM) представлена контекстним вікном, тоді як довготривала (LTM) організовується через векторні сховища embeddings. В агентних системах використовується також епізодична пам’ять – структурована інформація про виконані дії та їхні результати. Інженерна реалізація таких механізмів часто передбачає ведення

журналів виконання та використання засобів візуалізації історії взаємодії агента з середовищем.

Навички (skills) у GenAI-системах можуть розглядатися як програмно організовані модулі поведінки, що визначають способи виконання типових задач мовною моделлю або агентом. Такі модулі можуть включати інструкції, приклади використання, обмеження та опис доступних інструментів, що дозволяє повторно застосовувати їх у різних сценаріях генерації. Використання skills сприяє зниженню складності системного промпту, підвищує керованість поведінки агента та наближає GenAI-архітектури до модульних підходів класичного програмування. Подібні підходи до організації дій та використання інструментів у мовних моделях розглядаються в роботах, присвячених агентним архітектурам і взаємодії reasoning-та acting-компонентів [5].

Патерни безпеки та експлуатаційного контролю Guardrails, sandboxing і human-in-the-loop забезпечують передбачуваність поведінки моделей у критичних сценаріях [3]. Їх застосування супроводжується впровадженням інструментів моніторингу та візуалізації потоків генерації, що дозволяє інженерам виявляти небезпечні або некоректні сценарії роботи системи.

На відміну від традиційних програмних систем, де візуалізація переважно використовується для опису структурних залежностей компонентів, у GenAI-архітектурах вона виконує функцію інженерного контролю над динамікою генеративних процесів та управління якістю інтеграції мовних моделей у програмні продукти [3]. Одним із ключових підходів є візуалізація конвеєрів генерації, що відображає послідовність стадій обробки запиту: підготовку контексту, retrieval, генерацію відповіді, постобробку та валідацію. Представлення таких процесів у вигляді sequence-діаграм або графів виконання дозволяє інженерам виявляти надлишкові виклики моделі, оптимізувати latency і зменшувати витрати токенів. Подібні підходи застосовуються у практиці побудови LLM-pipeline-архітектур та систем оркестрації генеративних задач [4].

Візуалізація потоків знань у RAG-архітектурах, де retrieval-процеси моделюються як графи залежностей між запитом, retrieved-документами та

сформованими твердженнями. Це забезпечує можливість трасування походження відповідей моделі та спрощує виявлення некоректних або нерелевантних джерел контексту. Дослідження retrieval-augmented моделей показують, що прозорість джерел контексту є критичною умовою підвищення довіри до результатів генерації [2].

У агентних системах важливу роль відіграє візуалізація поведінкових сценаріїв, що може реалізовуватися через state-machine моделі або timeline-подання виконаних дій. Такі інструменти дозволяють аналізувати прийняття рішень агентом, оцінювати ефективність використання пам’яті та виявляти зациклення або неочікувані переходи між задачами. Подібні підходи застосовуються у фреймворках автономних агентів і систем планування дій [5].

Крім того, практичного значення набуває операційна візуалізація експлуатаційних характеристик GenAI-систем, зокрема dashboards контролю latency, throughput, вартості генерації та якості відповідей. На відміну від класичних систем моніторингу, ці показники безпосередньо впливають на семантичну якість результатів, що створює нові вимоги до інженерного аналізу продуктивності. Концепції observability у розподілених системах дедалі частіше адаптуються до задач моніторингу LLM-викликів та поведінки генеративних сервісів [6]. Разом із тим використання візуалізаційних підходів має обмеження. Надмірна деталізація графів виконання або retrieval-структур може ускладнювати інтерпретацію поведінки системи, а також збільшувати витрати на інструментальну підтримку. Це вимагає формування узагальнених моделей візуалізації, які поєднують інформативність і масштабованість.

У роботі проаналізовано інженерні патерни проєктування систем генеративного штучного інтелекту, що формують окремий шар взаємодії з недетермінованими когнітивними сервісами. Показано, що застосування промпт-орієнтованих підходів, механізмів керування контекстом, інтеграції зовнішніх знань і програмної організації пам’яті дозволяє підвищити передбачуваність поведінки моделей, оптимізувати використання обчислювальних ресурсів та забезпечити масштабованість GenAI-рішень у

виробничих середовищах. Обґрунтовано інженерну роль візуалізаційних підходів як інструменту аналізу та оптимізації генеративних процесів, що сприяє підвищенню прозорості архітектури, спрощує трасування сценаріїв взаємодії з моделлю та підтримує контроль експлуатаційних характеристик систем.

Список використаних джерел:

1. Wei J., Wang X., Schuurmans D., Bosma M., Ichter B., Xia F., Chi E. H., Le Q. V., Zhou D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models // NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing System. 2022. Vol. 35, Article No. 1800. P. 24824–24837.
2. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W.-t., Rocktäschel T., Riedel S., Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020. Vol. 33, Article No. 793. P. 9459–9474.
3. Subramaniam B., Fowler M. Emerging Patterns in Building GenAI Products // Martin Fowler. 2025. 25 Feb. URL: martinfowler.com/articles/gen-ai-patterns/ (дата звернення: 20.03.2026).
4. Lu Q., Zhu L., Xu X., Xing Z., Harrer S., Whittle J. Towards Responsible Generative AI: A Reference Architecture for Designing Foundation Model based Agents // IEEE Software. 2024. Volume 41, Issue 6. P. 91-100. URL: DOI: https://doi.org/10.1109/MS.2024.3406333.
5. Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., Cao Y. ReAct: Synergizing Reasoning and Acting in Language Models // The Eleventh International Conference on Learning Representations (ICLR 2023). URL: openreview.net/forum?id=WE_vluYUL-X (дата звернення: 20.03.2026).
6. Dong L., Lu Q., Zhu L. AgentOps: Enabling Observability of LLM Agents // arXiv. 2024. URL: arxiv.org/abs/2411.05285 (дата звернення: 20.03.2026).

УДК 004.415.2:004.8

ЗАСТОСУВАННЯ LLM ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ ТЕСТОВОЇ ДОКУМЕНТАЦІЇ

Кітораги В. О.
kitoragivioleta25@gmail.com
Черкаський державний фаховий бізнес-коледж
Подорошко Д. І.
м. Черкаси, Україна

У сучасній розробці програмних продуктів тестування та контроль якості займають одне з ключових місць. Проте спеціалісти з QA витрачають чималу частину робочого часу на підготовку різноманітної документації − зокрема, чек-листів, тест-кейсів, баг-репортів та тестових сценаріїв. Як зазначають фахівці, формування повноцінного набору тест-кейсів навіть для однієї вимоги може бути трудомістким процесом, що особливо відчутно для тих, хто лише починає працювати у цій сфері [3].

Розвиток великих мовних моделей створив передумови для часткової автоматизації подібних рутинних завдань. Проблема полягає в тому, що наявні на ринку інструменти здебільшого є універсальними та не враховують особливостей QA-процесів і специфіки їхньої документації [2].

У зв'язку з цим дана робота присвячена вивченню підходу до розробки вузькоспеціалізованого веб-застосунку, який на базі великих мовних моделей автоматично генерує три види QA-артефактів з дотриманням прийнятих у галузі стандартів оформлення.

Для обґрунтування необхідності розробки власного рішення було здійснено огляд наявних інструментів на ринку. Серед спеціалізованих платформ варто виділити TestRail та Zephyr Scale, які орієнтовані на зберігання й управління тест-кейсами, втім функції їх автоматичної генерації в них відсутні. Універсальні AI-інструменти − ChatGPT, GitHub Copilot, Claude − певною мірою справляються зі створенням тестової документації, однак якість результату залежить від багатьох змінних [2].

Зокрема, при однаково сформульованому запиті, але з відмінними вхідними даними, модель може видавати результати з різною структурою та наповненням. Це суттєво ускладнює досягнення однорідності документації. Окрім цього, кінцевий результат значно варіюється залежно від того, як саме сформульований запит і який контекст передував поточному діалогу.

Варто також зазначити, що мовні моделі не зберігають інформацію між окремими сесіями. Для отримання стабільних і узгоджених відповідей необхідно працювати в межах одного діалогу, оскільки при відкритті нового всі попередні уточнення втрачаються. Це знижує зручність використання при тривалій роботі, а безкоштовні версії AI-інструментів додатково обмежують кількість доступних запитів.

Нарешті, існує ризик генерації недостовірної інформації або хибної інтерпретації запиту користувача. У подібних ситуаціях виникає необхідність повторного формулювання та деталізації запиту.

Розроблений застосунок являє собою односторінковий веб-додаток, побудований на основі Next.js 16 у поєднанні з React 19 та TypeScript. Вибір такої архітектури обумовлений можливістю об'єднати клієнтську та серверну логіку в єдиній кодовій базі, що значно спрощує процес розгортання й усуває потребу в окремому серверному компоненті. Генерація артефактів здійснюється за допомогою моделі llama-3.3-70b-versatile, розгорнутої через платформу Groq Cloud. Завдяки апаратним LPU-прискорювачам швидкість обробки становить від 200 до 500 токенів на секунду [2].

Функціонал застосунку охоплює три окремі режими роботи. Перший орієнтований на формування тест-кейсів: користувач надає назву функціональної вимоги, її опис та критерії приймання, після чого система генерує від 10 до 18 структурованих записів, кожен з яких містить унікальний ідентифікатор, передумови, покрокові дії, очікувані результати, пріоритет у діапазоні P0–P3 та відповідні теги. Другий режим призначений для оформлення баг-репортів: довільний опис виявленого дефекту перетворюється на стандартизований документ із визначеним рівнем серйозності (S1–S4) та

рекомендаціями щодо додаткових матеріалів. Третій режим дозволяє генерувати тест-ідеї для API: на вхід подається HTTP-ендпоінт, а на виході формується перелік сценаріїв, що охоплює як позитивні, так і негативні випадки, а також готові curl-команди для їх виконання [2].

Для забезпечення стабільного отримання структурованих даних від мовної моделі реалізовано багаторівневий механізм обробки відповідей, що складається з чотирьох послідовних етапів: безпосереднє перетворення у JSON, вилучення вмісту з Markdown-блоків коду, пошук JSON-структури за допомогою регулярного виразу та посимвольний розбір рядка. У випадку, коли жоден із зазначених методів не дає результату, застосунок автоматично формує повторний запит із відповідним коригуючим повідомленням [2].

Якість генерованих документів значною мірою визначається тим, як побудовано системний промпт. У ньому зафіксовано перелік небажаних формулювань − зокрема, розмитих описів на кшталт «має з'явитися помилка» без вказівки конкретного тексту та елемента інтерфейсу − а також вимоги до кожного поля та еталонні приклади у форматі few-shot. Додавання двох зразкових тест-кейсів безпосередньо до промпту помітно підвищило точність очікуваних результатів: середня оцінка якості на однакових вхідних даних зросла з 6,5 до 8,5 за шкалою від 1 до 10.

З метою запобігання надмірному навантаженню на сервіс впроваджено механізм обмеження запитів − не більше десяти звернень на хвилину з однієї IP-адреси, реалізований через структуру Map у пам'яті процесу. Персональні дані користувача зберігаються виключно на стороні клієнта у localStorage браузера: система підтримує до 50 записів в історії та автоматично очищає сховище при досягненні порогу у 4 МБ, а також зберігає контекст поточного проєкту та чернетки заповнених форм.

Практичну перевірку застосунку було здійснено шляхом порівняння витрат часу на ручне створення QA-документації та з використанням розробленого інструменту. Результати показали суттєве скорочення часових витрат у всіх трьох режимах. Формування набору з 14 тест-кейсів для однієї

функції вручну потребує від 45 до 60 хвилин, тоді як із застосуванням асистента цей процес займає 5–8 хвилин, що відповідає економії близько 87%. Час на оформлення баг-репорту зменшився з 10–15 до 2–3 хвилин (приблизно 80%), а підготовка ідей для тестування API-ендпоінту − з 20–30 до 3–5 хвилин (близько 85%). Застосунок є загальнодоступним та розміщений за адресою https://qa-help-eight.vercel.app/assistant [2].

Окремо варто відзначити освітній потенціал інструменту. Для тих, хто лише знайомиться з QA-практиками й не має досвіду складання тест-кейсів, згенеровані артефакти можуть виконувати роль наочних зразків, що демонструють прийняті у галузі стандарти оформлення. Для більш досвідчених користувачів ті самі матеріали стають зручним орієнтиром або шаблоном, що дозволяє пришвидшити та впорядкувати процес створення документації.

Проведене дослідження підтверджує, що вузькоспеціалізований QA-асистент на базі LLM демонструє вищу ефективність порівняно з універсальними AI-інструментами. Ключовими перевагами є передбачуваність структури вихідних даних, відповідність галузевим стандартам оформлення та менша залежність якості результату від точності формулювання запиту з боку користувача.

Розроблений інструмент дозволяє скоротити час на підготовку тестової документації на 80–87% і водночас забезпечує її формальну стандартизацію. Серед перспективних напрямів подальшого вдосконалення − підключення до платформ управління тестуванням, зокрема TestRail та Jira, впровадження RAG-компонента для роботи з проєктною документацією та специфікаціями, а також розширення функціоналу у бік автоматичної генерації тестових скриптів на основі Playwright або Cypress.

Список використаних джерел:

1. Fan A., Gokkaya B., Harman M. та ін. Large Language Models for Software Engineering: Survey and Open Problems. arXiv:2310.03533. 2023. URL: https://arxiv.org/abs/2310.03533 (дата звернення: 17.03.2026).

2. Кітораги В. О. Програмний асистент для підтримки роботи тестувальників : кваліфікаційна робота. Черкаси : ЧДБК, 2026.
3. Myers G. J., Sandler C., Badgett T. The Art of Software Testing. 3rd ed. Hoboken : John Wiley & Sons, 2011. 240 p.
4. Wang J., Huang Y., Chen C. та ін. Software Testing with Large Language Model: Survey, Landscape, and Vision. arXiv:2307.07221. 2023. URL: https://arxiv.org/abs/2307.07221 (дата звернення: 17.03.2026).
5. Hnatushenko V. V., Pavlenko I. V. Використання генеративного штучного інтелекту в тестуванні програмного забезпечення. Системні технології. 2024. Вип. 2(151). С. 10–20.

УДК 004.4

ОПТИМІЗАЦІЯ ПРОЦЕСУ АВТОМАТИЗОВАНОГО WEB-ПАРСИНГУ ЗА ДОПОМОГОЮ БАГАТОПОТОКОВОСТІ

Прудиус В.М.
agyshii@gmail.com
Черкаський державний технологічний університет
Метелап В.В.
м. Черкаси, Україна

Сучасний етап розвитку інформаційних технологій характеризується експоненційним зростанням обсягів даних у мережі Інтернет. Для аналізу ринку, машинного навчання, агрегації новин та інших науково-дослідних завдань критично важливим є інструментарій автоматизованого збору інформації, відомий як web-парсинг (web scraping). Традиційний однопотоковий підхід до реалізації подібних алгоритмів має суттєвий недолік – він характеризується значними часовими витратами та низьким коефіцієнтом корисної дії апаратного забезпечення. Це пов'язано із проблемою синхронного блокування (I/O Bound операції), при якому програма, відправивши мережевий запит, зупиняє своє виконання і більшу частину часу просто очікує на відповідь від цільового сервера та завантаження HTML-документа, тоді як ресурси центрального процесора (CPU) залишаються абсолютно незадіяними [1].

Ефективним та раціональним вирішенням цієї проблеми є впровадження архітектури багатопотоковості (Multithreading) або паралельних обчислень. Багатопотоковий парсинг базується на принципі конкурентного виконання завдань, що дозволяє ініціювати десятки або сотні мережевих запитів одночасно у межах одного процесу. Поки один потік очікує на відповідь від сервера, операційна система перемикає контекст на інший потік, який у цей час може здійснювати парсинг вже отриманого DOM-дерева або відправляти новий запит.

Оптимізація процесу за допомогою пулу потоків (Thread Pool) дозволяє масштабувати швидкість збору даних практично лінійно до певної межі, яка визначається пропускною здатністю мережевого каналу. Порівняльну ефективність застосування різних підходів до автоматизованого збору інформації на масиві з тисячі web-сторінок наведено у таблиці (табл. 1). Як видно з даних, застосування багатопотоковості зменшує час виконання завдання майже на порядок.

Таблиця 1 – Порівняння швидкодії та ресурсоємності методів web-парсингу

№	Метод обробки даних	Час виконання (1000 сторінок)	Завантаження CPU	Споживання ОЗП
1	Однопотоковий (Синхронний)	320 с	До 5%	50 МБ
2	Багатопотоковий (10 потоків)	38 с	15-20%	120 МБ
3	Багатопотоковий (50 потоків)	12 с	40-50%	350 МБ
4	Асинхронний (Event Loop)	9 с	25-30%	90 МБ

Проте, практична реалізація багатопотокового парсингу зіштовхується із серйозними перешкодами. Висока інтенсивність мережевих запитів з однієї IP-адреси миттєво ідентифікується системами кіберзахисту цільового ресурсу (наприклад, Web Application Firewalls – WAF, Cloudflare або AWS Shield) як потенційна DDoS-атака або нелегітимна активність бота [2]. Це призводить до

появи CAPTCHA, тимчасового або перманентного блокування доступу. Для нівелювання цих ризиків розробникам доводиться проектувати складну розподілену архітектуру із залученням проміжних вузлів (рис. 1).

Варто зазначити, що для успішного уникнення блокувань недостатньо лише змінити IP-адресу. Системи антифрод-захисту аналізують патерни поведінки та частоту звернень. Тому критично важливим є впровадження механізмів штучних затримок (Throttling). Математично, безпечний час затримки між запитами, який гарантує неперевищення лімітів цільового сервера, можна розрахувати за формулою (1):

$$T_{безпеч.} = (N_{потоків} \times T_{відповіді}) / K_{допуск}, \quad (1)$$

де:

$T_{безпеч.}$ – необхідний інтервал між зверненнями (у секундах);

$N_{потоків}$ – кількість одночасно активних робочих потоків;

$T_{відповіді}$ – середній час очікування відповіді від сервера;

$K_{допуск}$ – емпіричний коефіцієнт толерантності сервера до навантаження (зазвичай від 0.5 до 0.8).

Рисунок 1 – Архітектурна схема взаємодії багатопотокового парсера із цільовим сервером через ротацію проксі-пулу

Отже, проектування оптимізованого багатопотокового парсера вимагає комплексного дотримання наступних інженерних практик:

- використання обмежених пулів потоків на базі черг завдань (Task Queues) замість неконтрольованого створення нових;
- обов'язкова та випадкова ротація HTTP-заголовків (User-Agent, Accept-Language) та проксі-серверів для кожного нового з'єднання;
- вирішення проблем синхронізації (Race Conditions) за допомогою потокобезпечних структур даних для збереження зібраної інформації без втрати її цілісності;
- впровадження системи експоненційної затримки (Exponential Backoff) для коректної обробки мережевих помилок (коди 429, 503) та таймаутів.

Список використаних джерел

1. Мітчелл Р. Скрапінг web-сайтів із використанням Python. Київ: Видавництво, 2021. 300 с.
2. Вилучено з https://liga.science/conferences.html ; 2020 ГО Молодіжна наукова ліга.

УДК 004.738.5

РОЗРОБКА ФРОНТЕНДУ ЗА ДОПОМОГОЮ РІЗНИХ ФРЕЙМВОРКІВ: ПЕРЕВАГИ, НЕДОЛІКИ ТА ЗАВДАННЯ

Стеценко Я. І.
yanastetforcomp@gmail.com
Черкаський державний фаховий бізнес-коледж
Подорошко Д. І.
м. Черкаси, Україна

Сучасна розробка фронтенду є фундаментальною складовою створення прикладних рішень, що відповідають високим вимогам користувачів щодо швидкодії, масштабованості та безпеки. Еволюція вебтехнологій призвела до домінування односторінкових додатків (SPA), які забезпечують миттєву реакцію

без перезавантаження сторінок. Ці рішення базуються на JavaScript та TypeScript. Вибір інструменту є стратегічним рішенням, що впливає на життєвий цикл продукту. Індустрія сформувала чітку трійку лідерів: React, Vue.js та Angular, кожен з яких має унікальну архітектуру та специфічні сфери застосування, які потребують детального технічного аналізу.

Технологія React, бібліотека для створення користувацьких інтерфейсів, підтримується Meta. Її ключова інновація – використання Віртуального DOM (Virtual DOM). React обчислює різницю між поточним та новим станом (алгоритм узгодження) і точково оновлює лише необхідні елементи реального DOM, що сприяє підвищенню продуктивності у складних інтерфейсах.

Суворий компонентний підхід дозволяє розбивати інтерфейси на дрібні, незалежні та придатні для повторного використання блоки коду. Це значно спрощує командну розробку, тестування та рефакторинг. Величезна спільнота та широка екосистема роблять React універсальним інструментом.

Серед недоліків: високий поріг входження через відсутність суворої стандартизації архітектури «з коробки» (розробникам доводиться самостійно обирати інструменти для маршрутизації та управління станом, наприклад, Redux або Zustand). Часті оновлення парадигм, зокрема перехід до функціональних компонентів з хуками, вимагають постійного навчання та адаптації існуючого коду. React використовують для створення інтерактивних дашбордів, розгалужених платформ електронної комерції, соціальних мереж та стрімінгових сервісів. Наприклад, під час розробки проєкту MusicHub компонентний підхід React дозволив команді ефективно реалізувати динамічний каталог нотного матеріалу з миттєвим пошуком та створити безперебійний процес оформлення підписки, забезпечивши високу стабільність платформи. [1]

Vue.js – це прогресивний фреймворк, розроблений для об'єднання найкращих архітектурних рішень та створення максимально зрозумілого та гнучкого продукту. Він характеризується надзвичайно адаптивною системою інтеграції, дозволяючи використовувати його як легку бібліотеку для

інтерактивності на окремих сторінках, або як повноцінний фреймворк для складних корпоративних додатків (із Vue Router та Pinia).

Найсильнішою стороною є реактивна система збору залежностей, яка працює практично непомітно для розробника, автоматично відстежуючи зміни в даних та миттєво оновлюючи інтерфейс. Розробники високо оцінюють парадигму однофайлових компонентів (JavaScript, HTML, CSS в єдиному файлі), що підвищує візуальну зрозумілість структури проєкту. Слабкою стороною традиційно виділяють дещо меншу популярність у великому корпоративному секторі порівняно з конкурентами, хоча ситуація швидко змінюється. Vue.js часто застосовується для розробки фронтенд-інтерфейсів сучасних фінансових сервісів та цифрових гаманців. Висока швидкість рендерингу, легкість у підтримці коду та зручність управління складним фінансовим станом дозволяють створювати надійні клієнтські рішення, які витримують значні навантаження. [2]

Angular – це комплексний фреймворк, який розробляється та підтримується корпорацією Google. На відміну від React та Vue, Angular пропонує філософію повної комплектації, містячи вбудовані стандартизовані інструменти для всіх можливих завдань веброзробки: від маршрутизації та роботи з формами до HTTP-запитів та глибокого модульного тестування. Базовою мовою програмування є TypeScript, що забезпечує сувору типізацію коду, використання об'єктно-орієнтованих патернів та виявлення архітектурних помилок ще на етапі компіляції.

Архітектура Angular суворо базується на концепціях модульності, сервісів та впровадження залежностей (Dependency Injection), що робить його ідеальним вибором для великих інженерних команд, де критично важливо дотримуватися єдиного суворого стилю написання коду. Angular характеризується складнішим порогом входження, оскільки для продуктивної роботи розробникам необхідно глибоко опанувати концепції реактивного програмування за допомогою бібліотеки RxJS. Монолітність та великий початковий розмір кінцевого пакета часто роблять його технічно невиправданим для невеликих проєктів. Основна

ніша застосування Angular – розробка масштабних корпоративних систем управління, складних CRM та банківських порталів, що мають високі вимоги до стабільності, прогнозованості та довготривалої підтримки. [3]

Окрім вибору безпосередньо фреймворку, сучасна розробка фронтенду вимагає врахування архітектури безпеки, зокрема, впровадження концепції нульової довіри (Zero Trust) на рівні клієнтського додатка. Фронтенд-розробники повинні забезпечувати надійне зберігання авторизаційних даних, правильне налаштування політик спільного використання ресурсів та безпечну обробку токенів сесій. Хоча сучасні версії фреймворків автоматично екранують дані, мінімізуючи ризики міжсайтового скриптингу (XSS), розвиток технологій штучного інтелекту стимулює появу нових загроз. Тому вибір сучасного та регулярно оновлюваного фреймворку є критичною необхідністю для забезпечення стабільності інформаційних систем та захисту даних користувачів.

Детальний технічний аналіз сучасного інструментарію веброзробки підтверджує, що вибір фронтенд-технології залежить від специфіки конкретного продукту, вимог до його довгострокового масштабування та наявного досвіду команди.

- React залишається найбільш універсальним та гнучким вибором для більшості динамічних платформ, де ключову роль відіграє насичена взаємодія з користувачем та швидкість відмальовування компонентів.
- Vue.js приваблює структурною елегантністю, відносно низьким порогом входження та простотою поступової інтеграції, що робить його затребуваним при створенні цифрових гаманців та легких комерційних сервісів.
- Angular міцно широко застосовується для побудови корпоративних інформаційних систем найвищої складності, де пріоритетами виступають сувора типізація, жорстка прогнозованість архітектури та наявність єдиного стандартизованого набору інструментів.

Практика розробки підтверджує, що грамотний та обґрунтований вибір технологічного стека дозволяє не лише значно оптимізувати процеси

програмування, але й гарантувати створення надійного, стійкого до сучасних кіберзагроз та максимально зручного інтерфейсу, що відповідає найвищим стандартам якості програмного забезпечення.

Список використаних джерел:

1. React – A JavaScript library for building user interfaces. Official Documentation. URL: https://react.dev/
2. MDN Web Docs: Front-end web developer. Mozilla Corporation. URL: https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer
3. OWASP Top 10 Client-Side Security Risks. Open Worldwide Application Security Project. URL: https://owasp.org/www-project-top-10-client-side-security-risks/
4. Web Security Guidelines. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/Security
5. Osmani A. Learning JavaScript Design Patterns: A JavaScript and React Developer's Guide. 2nd Edition. O'Reilly Media, 2023. 280 p.

УДК 004.42

ОСНОВНІ ПІДХОДИ ДО ФІЗИЧНОГО МОДЕЛЮВАННЯ У 3D ГРІ-ГОЛОВОЛОМЦІ «SPHERECAGE»

Соловйов І.С.
olegzaharov141@gmail.com
Черкаський державний фаховий бізнес-коледж
Подорошко Д. І.
м. Черкаси, Україна

Сучасний етап розвитку інформаційних технологій характеризується зростанням ролі індустрії відеоігор, які використовуються не лише для розваг, а й у освітніх та тренувальних цілях. Особливе місце посідають 3D-ігри, що забезпечують високий рівень занурення у віртуальне середовище завдяки реалістичному відображенню простору.

Ключовим аспектом, що впливає на реалістичність ігрового процесу, є фізичне моделювання. Саме воно відповідає за природну поведінку об'єктів,

їхню взаємодію та рух, що безпосередньо впливає на сприйняття гри користувачем.

Метою даної роботи є розробка простої 3D гри-головоломки, ігрова механіка якої базується на керуванні сферою для проходження лабіринту та досягнення кінцевої точки. Такий підхід дозволяє дослідити базові принципи фізичної взаємодії об'єктів у тривимірному просторі.

Для реалізації проєкту обрано ігровий рушій Unity, що зумовлено його функціональністю, інтуїтивно зрозумілим інтерфейсом, наявністю вбудованої системи фізики (Rigidbody) та широкою базою навчальних матеріалів, які спрощують процес розробки.

У ході виконання роботи було реалізовано систему керування віртуальною сферою за допомогою клавіатури. Рух об'єкта відбувається з урахуванням фізичних параметрів (сили, тертя, інерції), що забезпечує більш реалістичну поведінку порівняно з простим переміщенням. Лабіринт сформовано з базових геометричних примітивів, які виконують функцію статичних перешкод.

Експериментальним шляхом встановлено, що варіювання фізичних параметрів (швидкості руху, коефіцієнтів тертя) суттєво впливає на складність ігрового процесу. Це підкреслює необхідність ретельного налаштування цих характеристик для досягнення балансу між реалістичністю та зручністю керування.

Результатом роботи став функціональний прототип гри, який демонструє стабільну роботу механіки проходження лабіринту. Розроблений додаток підтверджує ефективність використання фізичного моделювання навіть у проєктах з простою логікою для підвищення якості взаємодії з користувачем.

Список використаних джерел:

1. Unity Documentation. URL: https://docs.unity.com/ (дата звернення: 20.03.2026).

2. Unity Manual Physics. URL: https://docs.unity3d.com/Manual/PhysicsSection.html (дата звернення: 20.03.2026).
3. Introduction to Game Development with Unity. Unity Learn. URL: https://learn.unity.com/ (дата звернення: 19.03.2026).
4. Не тільки для геймерів: Як Unity змінює різні індустрії. *Robot Dreams*. 2024. URL: https://robotdreams.cc/uk/blog/705-how-unity-became-a-tool-for-everything (дата звернення: 20.03.2026).

УДК 004.4

ВИКОРИСТАННЯ БАГАТОПОТОЧНОСТІ В СУЧАСНИХ ІГРОВИХ РУШІЯХ (НА ПРИКЛАДІ UNITY)

Різник О.М.
alex07082005m@gmail.com
Черкаський державний технологічний університет, Україна
Метелап В.В.
м. Черкаси, Україна

Сучасні ігрові застосунки належать до класу систем реального часу, для яких критичною є здатність обробляти великі обсяги даних із мінімальними затримками. Зі зростанням складності ігрових сцен, кількості об’єктів, штучного інтелекту та фізичних симуляцій, продуктивність однопотокових обчислень стає недостатньою. У зв’язку з цим актуальним є використання багатопоточності, що дозволяє ефективно задіяти багатоядерні процесори.

Метою даної роботи є аналіз застосування багатопоточності в сучасних ігрових рушіях на прикладі Unity, а також визначення основних підходів до розпаралелення обчислень.

Потік виконання (thread) є мінімальною одиницею виконання програми, яка може працювати паралельно з іншими потоками в межах одного процесу. Використання декількох потоків дозволяє одночасно виконувати незалежні частини програми, що є основою паралельних обчислень.

Слід розрізняти поняття конкурентності та паралелізму. Конкурентність означає здатність системи працювати з кількома задачами, тоді як паралелізм передбачає їх одночасне виконання на різних обчислювальних ядрах/процесорах.

Разом з перевагами багатопоточність створює низку проблем, серед яких:

- стани гонки (race conditions) – некоректний доступ до спільних даних;
- взаємне блокування (deadlock) – ситуація, коли потоки очікують один одного;
- накладні витрати синхронізації – зниження продуктивності через використання механізмів блокування.

Таким чином, ефективне використання багатопоточності потребує спеціальних архітектурних підходів.

Ігрові рушії реалізують так званий ігровий цикл (game loop), який включає обробку логіки, фізики та рендеринг. У класичному підході ці етапи виконуються послідовно в одному потоці, що створює вузьке місце (bottleneck).

Для підвищення продуктивності сучасні рушії використовують багатопоточність для розподілу задач:

- обробка штучного інтелекту;
- фізичні симуляції;
- анімація;
- завантаження ресурсів;
- обчислення великої кількості об’єктів.

Такий підхід дозволяє значно зменшити час виконання кадру та забезпечити стабільну частоту оновлення екрану.

Unity є одним із найпоширеніших ігрових рушіїв, який реалізує сучасні підходи до паралелізації через набір технологій, відомий як Data-Oriented Technology Stack (DOTS).

Unity має свій набір інструментів для оптимізації роботи додатків та прискорення їх роботи через паралельні процеси.

Unity Job System дозволяє розбивати обчислення на невеликі незалежні задачі (jobs), які можуть виконуватися паралельно на різних ядрах процесора. Такі задачі описуються як структури та виконуються через спеціальний планувальник.

Burst – це компілятор, що перетворює проміжний код .NET у високоефективний машинний код із використанням оптимізацій LLVM. Він призначений для роботи з Job System та дозволяє значно підвищити продуктивність CPU-обчислень.

Зокрема, розбиття задач на незалежні job-и та їх компіляція за допомогою Burst дозволяє «ефективно використовувати всі доступні ядра процесора».

ECS є архітектурним підходом, орієнтованим на дані. Він відокремлює дані (компоненти) від логіки (системи), що дозволяє обробляти великі масиви об’єктів більш ефективно.

Згідно з документацією Unity, ECS дозволяє:

- масштабувати симуляції до великої кількості об’єктів;
- ефективно використовувати ресурси процесора та пам’яті;
- підвищити продуктивність завдяки data-oriented підходу.

Також ECS працює у зв’язці з Job System та Burst Compiler, що забезпечує максимальну ефективність використання апаратних ресурсів.

Окрім багатопоточності, Unity підтримує асинхронні механізми (coroutines, async/await), які дозволяють виконувати операції без блокування основного потоку, зокрема:

- завантаження сцен;
- підвантаження ресурсів;
- мережеві запити.

Багатопоточність у Unity використовується для вирішення ряду практичних задач:

- обробка великої кількості NPC;
- процедурна генерація світу;

– фізичні симуляції;
– потокове завантаження контенту.

Застосування ECS дозволяє обробляти «великомасштабні симуляції та значну кількість сутностей» , що є критичним для сучасних ігор.

Попри переваги, багатопоточність у Unity має ряд обмежень:

– більшість API доступна лише з головного потоку;
– складність синхронізації між потоками;
– труднощі відлагодження;
– накладні витрати при надмірній паралелізації.

Крім того, неправильне використання потоків може призвести до зниження продуктивності замість її покращення.

Багатопоточність є ключовим інструментом підвищення продуктивності сучасних ігрових рушіїв. Вона дозволяє ефективно використовувати багатоядерні процесори та забезпечувати стабільну роботу ігор у реальному часі.

На прикладі Unity показано, що ефективна реалізація багатопоточності досягається не лише створенням потоків, а й використанням спеціалізованих архітектурних підходів, таких як Job System, Burst Compiler та ECS.

Таким чином, сучасні ігрові рушії переходять від класичних об’єктно-орієнтованих моделей до data-oriented підходів, що дозволяє досягти значного приросту продуктивності.

Список використаних джерел:

1. Unity Technologies. ECS for Unity. URL: https://unity.com/ecs (дата звернення: 24.04.2026).
2. Unity Technologies. Burst Compiler Documentation. URL: https://docs.unity3d.com (дата звернення: 24.04.2026).
3. Unity Technologies. Job System and ECS Tutorial. URL: https://learn.unity.com (дата звернення: 24.04.2026).
4. Unity Technologies. Data-Oriented Technology Stack (DOTS). URL: https://unity.com (дата звернення: 24.04.2026).

СЕКЦІЯ 3

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ГАЛУЗЕВИХ РІШЕННЯХ

УДК 004.77

МОЖЛИВОСТІ СТВОРЕННЯ ЦИФРОВИХ АРОМАТІВ ТА ЇХ ПЕРЕДАЧІ ЧЕРЕЗ ІНТЕРНЕТ

Білоголова П. Я.
polinabiloholova2110@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Сучасні технології вже дозволяють передавати звук, зображення та відео через інтернет. Але чи можливо передавати запахи? Уявімо світ, де ми можемо не тільки побачити фото кави чи парфумів, а й відчути їх аромат. Саме це намагається реалізувати технологія цифровізації запахів.

Цифровізація запаху – це процес захоплення, аналізу та перетворення аромату у цифрові дані.

Один із основних підходів – використання спеціальних сенсорів (так званих «електронних носів»), які аналізують хімічний склад запаху та перетворюють його у цифровий код. Далі штучний інтелект обробляє ці дані та створює модель аромату [1].

Наприклад, компанія Osmo використала метод газової хроматографії-мас-спектрометрії, щоб розкласти запах на молекули, перетворити їх у дані та навіть відтворити аромат сливи [4].

Інший спосіб – генерація запахів. Вчені створюють пристрої, які можуть відтворювати аромати за допомогою спеціальних матеріалів. Наприклад, у нових VR-пристроях використовуються пластини з ароматизованого парафіну, які нагріваються і швидко виділяють запах. Інтенсивність контролюється температурою [2].

Також існує підхід мікрокапсуляції – коли молекули запаху «запаковуються» у маленькі капсули, які можуть вивільняти аромат при нагріванні або інших умовах.

Ще більш незвичайний метод – стимуляція нюхових нервів. У деяких експериментах вчені передавали електричні сигнали безпосередньо до нюхових рецепторів, і люди відчували різні запахи без реальних ароматичних речовин [3].

Після того як запах перетворено у цифровий код, його можна передавати через інтернет так само, як інші дані.

Процес виглядає так:

- аналіз запаху;
- створення цифрового коду;
- передача через мережу;
- відтворення на іншому пристрої.

Дослідники компанії Osmo вперше змогли передати запах на відстань: дані про аромат завантажувалися в хмару, передавалися на інший пристрій і там відтворювалися [4].

У сфері віртуальної реальності вже створюються бездротові пристрої, які можуть отримувати сигнал і генерувати запахи в реальному часі [2]. Це дозволяє зробити віртуальний світ більш реалістичним.

Технологія цифрових запахів має багато можливих застосувань:

- у віртуальній реальності та іграх;
- у маркетингу (передача запаху товарів);
- у медицині (діагностика та відновлення нюху);
- у безпеці (виявлення диму або небезпечних газів);
- у промисловості та сільському господарстві.

Також “електронні носи” вже використовуються для контролю якості продуктів і повітря [1].

Однак існують і проблеми:

- складність точного відтворення запахів;
- відсутність єдиного стандарту;
- висока вартість технологій;
- різне сприйняття запахів людьми.

Отже, цифровізація запахів і їх передача через інтернет – це новий і перспективний напрям. Уже сьогодні існують технології, які дозволяють аналізувати, кодувати та відтворювати аромати.

Хоча ця сфера ще розвивається, у майбутньому вона може стати такою ж звичною, як передача звуку чи відео, і значно змінити наш досвід взаємодії з цифровим світом.

Список використаних джерел:

1. Що таке цифровізація запахів: проривний бізнес-інструмент чи черговий хайп. URL: https://psm7.com/uk/technology/kak-cifrovizaciya-zapaxov-mozhet-izmenit-mir-razrabotka-i-ispolzovanie-texnologii.html
2. Вчені створили пристрій, що дозволяє відчувати запах у віртуальній реальності. URL: https://www.imena.ua/blog/device-that-allows-you-to-smell-in-virtual-reality/
3. Учені винайшли спосіб передачі віртуальних запахів: як це працює. URL: https://life.pravda.com.ua/society/2018/10/19/233726/
4. Перша у світі телепортація запаху. URL: https://hackyourmom.com/novyny/persha-u-sviti-teleportacziya-zapahu/

УДК 004.8:347.94

ШТУЧНИЙ ІНТЕЛЕКТ ДЛЯ АНАЛІЗУ ДОКАЗІВ ТА ОПТИМІЗАЦІЇ СУДОВИХ РІШЕНЬ

Віхренко О.В.
oleksandrvikhrenko@gmail.com
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м. Черкаси, Україна

Розвиток цифрових технологій у XXI столітті зумовив поступову трансформацію підходів до організації судочинства. Зростання кількості судових справ, ускладнення правовідносин та необхідність обробки значних обсягів інформації актуалізують питання впровадження інструментів, здатних

підвищити ефективність діяльності судової системи. У цьому контексті особливу увагу привертає штучний інтелект, який розглядається не лише як технологічна інновація, а як потенційний елемент модернізації правосуддя.

Сучасні системи на основі штучного інтелекту демонструють здатність працювати з великими масивами даних, аналізувати тексти, виявляти закономірності та формувати узагальнені висновки. У сфері судочинства це відкриває можливості для автоматизації окремих етапів розгляду справ, передусім тих, що пов'язані з пошуком і опрацюванням інформації. Зокрема, йдеться про аналіз судової практики, систематизацію нормативно-правових актів та підтримку підготовки процесуальних документів.

Окреме місце займає застосування штучного інтелекту під час аналізу доказів. Цей етап є одним із найбільш складних у судовому процесі, оскільки передбачає не лише формальне опрацювання матеріалів, але й встановлення логічних зв'язків між ними. Алгоритмічні моделі здатні значно пришвидшити цей процес, обробляючи великі обсяги текстових і структурованих даних, виявляючи суперечності або повторювані елементи у показаннях. Це дозволяє зменшити навантаження на суддю та підвищити якість попереднього аналізу справи.

Разом із тим застосування таких технологій не є однозначно позитивним. Практика свідчить, що результати, отримані за допомогою штучного інтелекту, можуть відрізнятися залежно від умов використання, що ставить під сумнів їхню стабільність. У сфері правосуддя, де важливу роль відіграє послідовність і передбачуваність рішень, це створює додаткові виклики. Крім того, існує ризик використання недостовірної інформації, сформованої алгоритмами, що може вплинути на зміст судових документів.

Показовим є те, що в українській судовій практиці вже зафіксовано випадки, коли використання штучного інтелекту стало підставою для перегляду судових рішень. Зокрема, було встановлено, що під час підготовки тексту вироку використовувалися згенеровані матеріали, що не відповідало вимогам процесуального законодавства. Така ситуація підкреслює необхідність

обережного та усвідомленого підходу до інтеграції нових технологій у правову сферу.

Перспективним напрямом розвитку є впровадження штучного інтелекту в електронні судові сервіси. В умовах цифровізації державних послуг система електронного суду може стати платформою для використання інтелектуальних інструментів, які виконуватимуть консультативні функції. Йдеться про допомогу користувачам у формуванні процесуальних документів, орієнтацію у судовій практиці та пояснення окремих процедурних аспектів. Такий підхід здатний підвищити доступність правосуддя та спростити взаємодію громадян із судовою системою.

Водночас подібні зміни можуть вплинути на традиційні моделі правничої діяльності. Часткова автоматизація юридичних процесів потенційно знижує потребу у виконанні рутинних завдань, які раніше здійснювалися юристами або адвокатами. Це не означає повного витіснення професії, однак свідчить про необхідність її трансформації та адаптації до нових умов.

Особливо дискусійним залишається питання можливості використання штучного інтелекту безпосередньо для прийняття судових рішень. Прихильники такої ідеї звертають увагу на потенційну неупередженість алгоритмів, які не мають власних переконань або емоцій і діють виключно на основі доступних даних. З цієї точки зору штучний інтелект може розглядатися як засіб мінімізації людського фактора.

Разом із тим подібний підхід має суттєві обмеження. Судове рішення є не лише результатом логічного аналізу, але й актом правозастосування, який передбачає врахування конкретних обставин справи, інтерпретацію норм права та дотримання принципів справедливості. У цьому контексті повна передача функцій судді алгоритмічній системі виглядає передчасною, оскільки сучасні технології не здатні відтворити комплексність людського мислення.

Не менш важливим аспектом є питання захисту персональних даних. Використання цифрових інструментів у судочинстві пов’язане з обробкою значних обсягів конфіденційної інформації, що вимагає дотримання відповідних

стандартів безпеки. У сучасних умовах це питання набуває особливої актуальності, оскільки витік або неправомірне використання даних може мати серйозні наслідки.

На нормативному рівні в Україні вже здійснюються спроби врегулювання використання штучного інтелекту у правовій сфері. Розробляються рекомендації щодо відповідального застосування таких технологій, а також формуються внутрішні політики їх використання у судових органах. Це свідчить про поступове усвідомлення необхідності створення чітких правил інтеграції інновацій у систему правосуддя.

Таким чином, штучний інтелект виступає важливим інструментом, здатним підвищити ефективність аналізу доказів та оптимізувати окремі процеси прийняття судових рішень. Його впровадження відкриває нові можливості для розвитку судової системи, проте водночас потребує зваженого підходу, що враховує як технологічні переваги, так і потенційні ризики. Найбільш доцільною на сучасному етапі є модель, за якої штучний інтелект використовується як допоміжний засіб, зберігаючи ключову роль людини у здійсненні правосуддя.

Список використаних джерел

1. Штучний інтелект у правосудді – допомога чи загроза судовому розсуду. Верховний Суд. URL: https://supreme.court.gov.ua/supreme/pres-centr/news/1987559/
2. Штучний інтелект у правосудді – допомога чи загроза судовому розсуду. Верховний Суд. URL: https://supreme.court.gov.ua/supreme/pres-centr/news/1987559/
3. ШІ в судочинстві: від ідеї до реалій. Ternopil Live. URL: https://ternopillive.com.ua/shi-v-sudochinstvi-vid-ide%d1%97-do-realij/
4. Про захист персональних даних: Закон України № 2297-VI. URL: https://zakon.rada.gov.ua/go/2297-17
5. European Ethical Charter on the use of artificial intelligence in judicial systems. CEPEJ, 2018.
6. Regulation (EU) 2024/1689 (Artificial Intelligence Act).

УДК 004.738.5:630

РОЗРОБКА ІОТ-СИСТЕМИ МОНІТОРИНГУ СТАНУ ЛІСОВОГО ГОСПОДАРСТВА

Яценко М. С.,
matviy829@gmail.com,
Черкаський державний фаховий бізнес-коледж
Ночевнов Д.П.
м. Черкаси, Україна

Лісове господарство потребує оперативного контролю за пожежною небезпекою, змінами мікроклімату, станом насаджень і виникненням надзвичайних подій. Наразі Звенигородське лісове господарство використовує базові елементи цифровізації, основою яких є трирівневий відеонагляд, системи електронного обліку деревини та біржові аналітичні платформи [1]. Проте існуючий стан моніторингу критично залежить від людського фактора, оскільки операторам доводиться безперервно і вручну стежити за екранами відеокамер, що призводить до швидкої втоми та ризику пропустити ранні ознаки задимлення [2]. Традиційні форми спостереження, включаючи супутниковий моніторинг із його затримками та обльоти безпілотниками з їхніми суворими обмеженнями щодо ємності батарей, не завжди забезпечують швидке виявлення небезпечних змін.

Для розв’язання цих проблем пропонується перехід від епізодичних спостережень до автоматизованого збору та передавання даних у режимі, наближеному до реального часу, на основі технологій Інтернету речей. Апаратна інфраструктура запропонованого комплексу спирається на концепцію граничних обчислень (Edge Computing), що дозволяє суттєво оптимізувати мережевий трафік та зменшити навантаження на центральний сервер [3]. Роль локального концентратора та обчислювального вузла виконує одноплатний мікрокомп'ютер сімейства Raspberry Pi. Замість безперервної трансляції важкого відеопотоку на центральний сервер, цей пристрій здійснює локальний аналіз кадрів з підключених камер за допомогою алгоритмів машинного зору та легких

нейромереж [4]. Це дозволяє автоматизовано виявляти патерни задимлення безпосередньо на місці спостереження.

Налаштування комунікаційного середовища має базуватись на комбінованому використанні протоколів передачі даних. Збір первинної телеметрії (температури, вологості, рівня вуглекислого газу) здійснюється за допомогою сенсорних модулів, які передають дані через енергоефективну мережу стандарту LoRaWAN. Ця технологія забезпечує стійкий зв'язок на відстані кількох кілометрів в умовах щільної лісової забудови при надзвичайно низькому енергоспоживанні від автономних елементів живлення [5]. У разі позитивного спрацювання нейромережевого детектора на базі Raspberry Pi, система автоматично активуватиме спрямовані Wi-Fi мости, через які на пульт чергового лісництва миттєво передається сигнал тривоги та фрагмент відеозапису з фіксацією інциденту.

Для підтвердження технічної доцільності проєкту було розроблено математичну модель енергоспоживання автономного сенсорного вузла:

$$I_{avg} = \frac{I_{sleep} \cdot t_{sleep} + I_{meas} \cdot t_{meas} + I_{tx} \cdot t_{tx}}{T_{cycle}}$$, де

I_{avg} – середній струм споживання за один цикл;

I_{sleep} – струм під час режиму сну;

t_{sleep} – тривалість режиму сну;

T_{cycle} – загальний час циклу в секундах;

t_{meas} – час вимірювання;

I_{meas} – струм вимірювання;

t_{tx} – час передачі;

I_{tx} – струм передачі.

Було визначено, що протягом базового десятихвилинного інтервалу мікроконтролер переважно перебуває в режимі сну зі струмом споживання 0,1 мА. Фаза активного вимірювання та обробки даних триває 8 секунд і потребує

струму 15 мА, а передача пакету через LoRa-канал займає 2 секунди при піковому струмі 120 мА. За розрахунками середній струм споживання за один цикл становить близько 0,70 мА, що при використанні акумулятора ємністю 3000 мА·год забезпечує автономність вузла близько 178 діб.

Для забезпечення повної енергонезалежності апаратної інфраструктури в умовах щільної лісової забудови живлення низькопотужних кліматичних сенсорних вузлів доцільно реалізовувати на базі компактних аморфних сонячних панелей потужністю 0,5–2 Вт, які демонструють високу ефективність генерації навіть за умов розсіяного світла та часткового затінення під кронами дерев. Водночас для стабільної роботи локальних шлюзів на базі мікрокомп'ютерів Raspberry Pi, які виконують ресурсоємні завдання граничних обчислень та локального аналізу кадрів, необхідно використовувати монокристалічні сонячні панелі потужністю 15–30 Вт у поєднанні з MPPT-контролерами заряду для максимізації відбору енергії. Зважаючи на значні сезонні перепади температур у лісових масивах, накопичення згенерованої енергії для обох типів пристроїв оптимально здійснювати за допомогою літій-залізо-фосфатних (LiFePO4) акумуляторів, що гарантує безпеку експлуатації, збільшену кількість циклів заряду-розряду та загальну довговічність системи без необхідності частого технічного обслуговування [6].

Для захисту децентралізованої мережі, що працює у польових умовах і використовує бездротові канали, на локальних шлюзах доцільно розгорнути систему IDS/IPS на базі Snort, яка аналізуватиме мережевий трафік у реальному часі та блокуватиме спроби несанкціонованого доступу до телеметрії й налаштувань камер.

Практичне значення одержаних результатів полягає в тому, що запропоновані інженерні та програмні рішення формують захищену децентралізовану екосистему, здатну стабільно функціонувати в складних умовах лісових масивів. Впровадження такого комплексу дозволить скоротити час виявлення небезпечних змін у лісовому середовищі та зменшити

навантаження на персонал за рахунок переходу до подієво-орієнтованого моніторингу.

Список використаних джерел:

1. Букша І. Ф., Пастернак В. П., Пивовар Т. С. Рекомендації щодо розбудови державної системи моніторингу лісів України. Харків : УкрНДІЛГА, 2019. 35 с.
2. План ведення господарства (план лісоуправління) Звенигородського надлісництва філії «Центральний лісовий офіс» ДП «Ліси України» на 2026 рік. м. Звенигородка, 2025. 76 с.
3. Zhao M., Ye R.-J., Chen S.-T., Chen Y.-C., Chen Z.-Y. Realization of Forest Internet of Things Using Wireless Network Communication Technology of Low-Power Wide-Area Network. Sensors. 2023. Vol. 23, No. 10. Art. 4809. DOI: 10.3390/s23104809.
4. Yick J., Mukherjee B., Ghosal D. Wireless Sensor Network Survey. Computer Networks. 2008. Vol. 52, No. 12. P. 2292–2330. DOI: 10.1016/j.comnet.2008.04.002.
5. Bouguera T., Diouris J.-F., Chaillout J.-J., Jaouadi R., Andrieux G. Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN. Sensors. 2018. Vol. 18, No. 7. Art. 2104. DOI: 10.3390/s18072104.
6. Sofianidis I., Konstantakos V., Nikolaidis S. Reducing Energy Consumption in Embedded Systems Applications. Technologies. 2025. Vol. 13, No. 2. Art. 82. DOI: 10.3390/technologies13020082.

УДК 004.312:378.147

ОСОБЛИВОСТІ СТВОРЕННЯ НАВЧАЛЬНОГО АПАРАТНОГО ТРЕНАЖЕРА МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ

Кріт В.Є.
vitalikkrit5@gmail.com
Черкаський державний фаховий бізнес-коледж
Фальченко Н.Г.
м.Черкаси, Україна

Процес підготовки майбутніх інженерів у сфері інформаційних технологій неможливий без глибокого засвоєння принципів функціонування мікропроцесорної техніки. Сучасні вбудовані системи вимагають від розробника знань не лише у високорівневому програмуванні, а й у розумінні фізичних процесів, що відбуваються на рівні окремих регістрів та периферійних модулів мікроконтролера. Проте, пряме використання складного лабораторного обладнання на початкових етапах навчання може бути неефективним через високу вартість та ризики виходу заліза з ладу. Актуальність даної роботи зумовлена необхідністю створення універсального навчального тренажера, який би поєднував у собі етап безпечного віртуального моделювання та етап фізичної реалізації на базі платформи Arduino [1].

Важливою частиною дослідження є використання середовища Tinkercad як первинної ланки розробки. Ця платформа дозволяє проводити «м’яке» тестування електронних схем. В ході розробки тренажера в середовищі було реалізовано повноцінний цифровий макет, що включає мікроконтролер ATmega328P, систему індикації та блоки введення даних.

Симуляція дозволяє детально проаналізувати наступні аспекти:

- логіку роботи внутрішніх підтягуючих резисторів (PULL-UP) на цифрових входах;
- коректність адресації пристроїв на шині I2C при підключенні LCD-дисплея;
- стабільність роботи програмних циклів при одночасній обробці декількох переривань. Використання Tinkercad значно спрощує процес

налагодження (debugging), оскільки дозволяє відстежувати значення на виводах контролера в інтерактивному режимі без використання складного вимірювального обладнання [2].

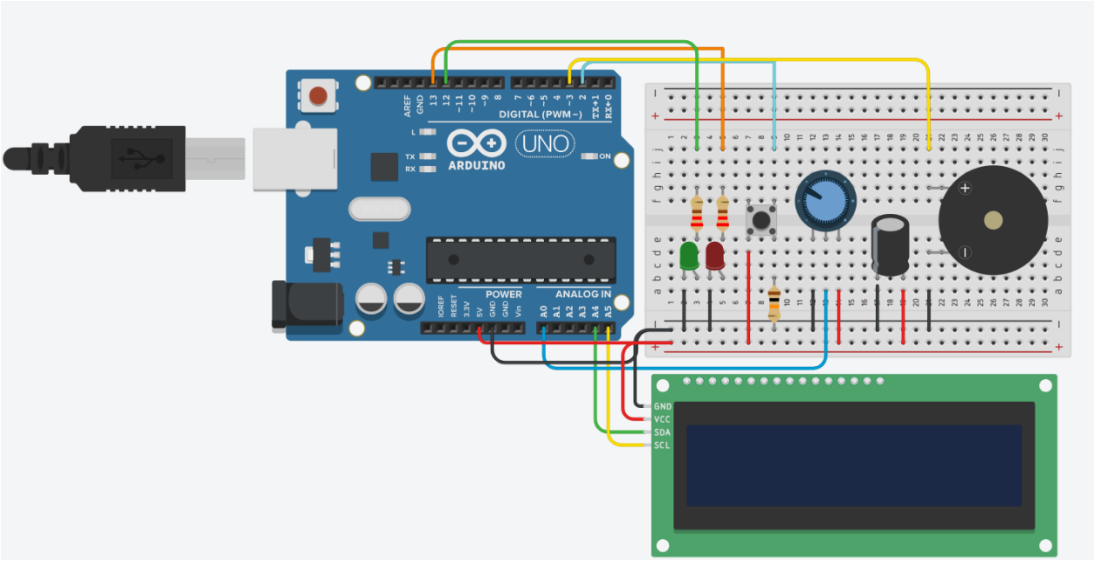


Рисунок 1 – Цифровий макет

Фізична частина тренажера побудована за модульним принципом на платформі Arduino Uno, де основною перевагою такої архітектури виступає її висока масштабованість. До складу розробленого пристрою інтегровано модуль візуалізації у вигляді 16-символьного дворядкового дисплея, який використовується для відображення поточних значень датчиків та виводу діагностичних повідомлень системи. Сенсорний блок тренажера включає потенціометр (замість нього можна використовувати інші типи датчиків), призначений для імітації зміни вхідної напруги, що дозволяє студентам детально вивчити роботу 10-бітного аналого-цифрового перетворювача (АЦП), де вхідна напруга в діапазоні 0–5В трансформується у відповідне цифрове значення від 0 до 1023 [3]. Крім того, систему доповнено керуючим блоком, що складається з набору кнопок, на яких практично відпрацьовується логіка обробки зовнішніх подій. Під час роботи з цим блоком студенти безпосередньо стикаються з фізичною проблемою «брязкоту» контактів (bounce effect), подолання якої

потребує реалізації алгоритмів програмних затримок або методів фільтрації сигналів.

При розробці тренажера особлива увага приділялася дотриманню стандартів передачі та обробки інформації. Згідно з вимогами до автоматизованих систем, програмне забезпечення тренажера має бути структурованим та забезпечувати надійне зчитування даних [4]. Використання стандартного протоколу I2C для зв'язку з дисплеєм дозволяє вивчати механізми синхронізації та адресації, що є фундаментальними для сучасної комп'ютерної інженерії. Це забезпечує сумісність розробленого тренажера з широким спектром промислових датчиків та виконавчих механізмів, що відповідають сучасним технічним регламентам.

Програмний комплекс розроблено на мові C++. Основна логіка базується на циклічному опитуванні станів (polling) та обробці переривань. Для підвищення точності роботи системи використано вбудовані таймери-лічильники. Модульна структура коду дозволяє студенту самостійно модифікувати окремі функції (наприклад, змінити поріг спрацювання датчика), не порушуючи роботу всієї системи.

Створений апаратний тренажер є повноцінним навчальним комплексом. Застосування тандему «Tinkercad – Arduino» дозволяє отримати комплексний досвід: від створення віртуальної схеми до роботи з реальними компонентами. Такий підхід підвищує безпеку навчання та стимулює інтерес до проектування вбудованих систем.

Список використаних джерел:

1. Гніденко М. П. Мікропроцесорні системи : підручник. Київ : НУВГП, 2021. 250 с.
2. Tinkercad: Circuits Release Notes and Documentation. URL: https://www.tinkercad.com/
3. Arduino Language Reference: Technical Manual. URL: https://www.arduino.cc/reference/en/

4. ДСТУ ISO/IEC 2382:2017. Інформаційні технології. Словник термінів. Київ: ДП «УкрНДНЦ», 2017. 110 с.

УДК 004.8

ВИКОРИСТАННЯ API В СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Ващенко В. В.
Vaschenkovladislav1p@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В. Ю.
м. Черкаси, Україна

У сучасному цифровому середовищі зростає потреба у взаємодії між різними програмними системами та сервісами. Одним із ключових інструментів такої взаємодії є API (Application Programming Interface), який забезпечує обмін даними між програмами та спрощує процес розробки програмного забезпечення. Використання API дозволяє інтегрувати сторонні сервіси, підвищувати ефективність створення програмних продуктів і скорочувати витрати часу на розробку. На рисунку 1 поданий принцип взаємодії клієнта і сервера через API.

API забезпечує взаємодію між клієнтською частиною додатка та сервером і базується на моделі запит відповідь, у межах якої клієнт надсилає запит, а сервер повертає результат у вигляді структурованих даних, найчастіше у форматі JSON. Найбільш поширеним підходом є REST, який використовує HTTP протокол для роботи з даними.

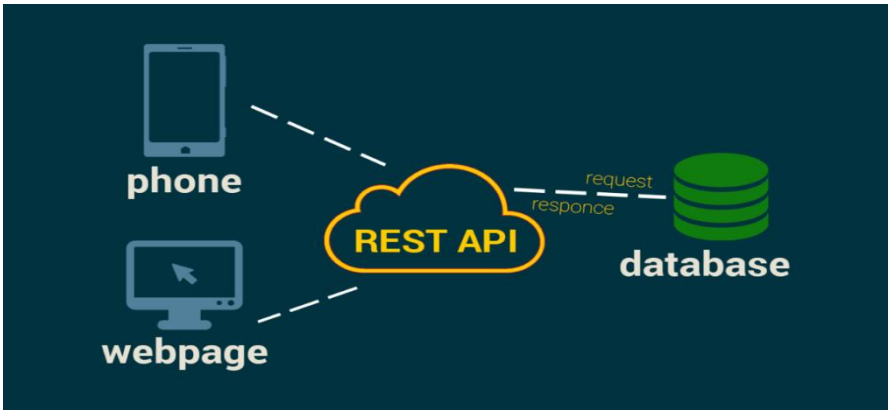


Рисунок 1 – Принцип взаємодії клієнта і сервера через API

Основними методами є GET для отримання інформації, POST для створення, PUT для оновлення та DELETE для видалення, що забезпечує простоту та ефективність взаємодії між системами. Окрім REST застосовуються також SOAP та GraphQL, де SOAP забезпечує високий рівень безпеки, а GraphQL дозволяє отримувати лише необхідні дані, зменшуючи обсяг переданої інформації. Також використовується WebSocket для обміну даними у режимі реального часу.

API широко застосовуються у веброзробці, мобільних додатках і хмарних сервісах, забезпечуючи інтеграцію різних функцій та ресурсів. Важливим аспектом є безпека, яка реалізується за допомогою токенів доступу, API ключів та шифрування даних, що дозволяє захистити інформацію від несанкціонованого доступу.

Таблиця 1 – Порівняння типів API

№	Тип API	Особливості	Переваги	Недоліки
1	REST	Використовує HTTP методи	Простота, популярність	Надлишкові дані
2	SOAP	Протокол на основі XML	Висока безпека	Складність реалізації
3	GraphQL	Запити тільки потрібних даних	Гнучкість, оптимізація	Складніша настройка
4	WebSocket	Постійне з’єднання	Реальний час	Вища складність підтримки

API є невід’ємним і надзвичайно важливим елементом сучасних інформаційних систем, оскільки вони забезпечують ефективну та надійну взаємодію між різними програмними компонентами та програмами. Завдяки використанню API розробники отримують можливість легко інтегрувати сторонні сервіси та функціональні модулі, що значно спрощує процес розробки програмного забезпечення. Крім того, API дозволяють створювати гнучкі, масштабовані та адаптивні програмні рішення, які легко підлаштовуються під змінні потреби користувачів та бізнес-логіку. Використання сучасних API сприяє підвищенню ефективності командної роботи, прискорює впровадження нових

функцій і забезпечує сумісність між різними платформами та технологічними стеком. Таким чином, API виступають ключовим механізмом для оптимізації розробки, інтеграції і підтримки сучасних цифрових систем, що робить їх незамінними у будь-якому програмному середовищі.

Список використаних джерел:

1. Fielding R. T., et al. REST API Design Rulebook. O'Reilly Media, 2011. URL: https://www.oreilly.com/library/view/rest-api-design/9781449317904/ (дата звернення: 16.03.2026).
2. OpenAPI Initiative. OpenAPI Specification. URL: https://www.openapis.org/what-is-openapi (дата звернення: 16.03.2026).
3. Jin B., Sahni S., Shevat A. Designing Web APIs: Building APIs That Developers Love. O'Reilly Media, 2018. URL: https://www.manning.com/books/the-design-of-web-apis-second-edition (дата звернення: 16.03.2026).
4. Peldszus S., et al. Developer Perspectives on REST API Usability: A Study of REST API Guidelines. arXiv, 2026. URL: https://arxiv.org/abs/2601.16705 (дата звернення: 16.03.2026).
5. Richardson L. Richardson Maturity Model. URL: https://martinfowler.com/articles/richardsonMaturityModel.html (дата звернення: 16.03.2026).

УДК 004.9

БІОМЕТРІЯ ЕМОЦІЙ: РОЗПІЗНАВАННЯ ПОЧУТТІВ ЧЕРЕЗ ТЕХНОЛОГІЇ

Оношко А. Ю.
anastasiaonosko10@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Емоції протягом століть вважалися невловимою частиною людської природи, однак у сучасну епоху цифрової трансформації вони перетворюються на дані, що можуть бути виміряні та проаналізовані. Розвиток технологій

штучного інтелекту сприяв активному впровадженню систем розпізнавання емоцій у різні сфери життя, зокрема медицину, бізнес і безпеку. Водночас це породжує низку етичних і правових викликів, пов'язаних із конфіденційністю та можливим зловживанням чутливими даними [2].

Системи розпізнавання емоцій, відомі як Emotion AI або Facial Emotion Recognition (FER), базуються на складному мультимодальному аналізі [3]. Для нас, людей, розуміння стану співрозмовника є природним процесом, який ми здійснюємо миттєво. Машина ж потребує складних математичних обчислень. Процес починається з детекції обличчя за допомогою будь-яких доступних сенсорів - від вуличних камер спостереження до фронтальних модулів наших смартфонів. Після цього алгоритм відокремлює ключові точки обличчя, так звані landmarks. Це специфічні координати на бровах, повіках, носі та губах. Саме відстеження руху цих точок і їх зіставлення з величезними базами даних дозволяє штучному інтелекту визначити тип емоції та її інтенсивність.

Більше того, сучасні нейромережі здатні фіксувати так звані мікроекспресії. Це ледь помітні зміни міміки, які тривають лише частку секунди. Людина не здатна свідомо їх контролювати, тому вони вважаються найбільш правдивими індикаторами внутрішнього стану. Саме на цій технології базуються сучасні детектори брехні нового покоління та системи безпеки на стратегічних об'єктах [1].

Окремим і надзвичайно перспективним напрямом є використання FER-систем у медицині. В галузі психіатрії та клінічної психології такі технології виступають об'єктивним інструментом оцінки, який не залежить від суб'єктивного сприйняття лікаря або бажання пацієнта приховати свої симптоми. Алгоритми здатні розпізнавати ранні маркери депресії, суїцидальних нахилів та навіть специфічну поведінку, характерні для розладів аутистичного спектра. У стаціонарних умовах ШІ може здійснювати безперервний моніторинг пацієнтів, оцінюючи їхню реакцію на медикаменти чи загальне задоволення умовами лікування. Для людей з обмеженими можливостями комунікації така система стає важливим інструментом взаємодії [3].

Інтеграція розпізнавання емоцій в автомобільну промисловість обіцяє кардинально змінити статистику дорожньо-транспортних пригод. Системи моніторингу аналізують стан водія та можуть вживати активних заходів для запобігання небезпеці.

У сфері управління персоналом технології розпізнавання емоцій допомагають оцінити стресостійкість кандидатів та їхню мотивацію, що сприяє формуванню ефективних команд.

Сфера правопорядку використовує Emotion AI для аналізу поведінки людей і запобігання загрозам, формуючи концепцію «розумного міста» [1].

Проте масове впровадження цих технологій породжує серйозні етичні та правові проблеми. Основним викликом є загроза порушення права на приватність і конфіденційність емоційних даних. Виникає ризик створення систем тотального контролю та маніпуляції суспільством.

Юристи наголошують, що втручання в емоційну сферу без згоди людини може порушувати її фундаментальні права. Крім того, помилки алгоритмів можуть призводити до дискримінації та неправильних рішень, що підкреслює необхідність правового регулювання [2].

Технології розпізнавання емоцій відкривають значні можливості для розвитку різних галузей, проте їх використання потребує обережного та відповідального підходу. Забезпечення балансу між технологічним прогресом і дотриманням прав людини є ключовою умовою їх ефективного впровадження. Необхідним є також удосконалення правового регулювання та підвищення точності алгоритмів для мінімізації ризиків [3].

Список використаних джерел:

1. The Page. Кому та навіщо потрібні технології розпізнавання емоцій. URL: https://thepage.ua/ua/it/komu-ta-navisho-potribni-tehnologiyi-rozpiznavannya-emocij (дата звернення: 18.03.2026).

2. Юргазета. Конституційно-правові проблеми використання ШІ в системах розпізнавання емоцій. URL: https://yur-gazeta.com (дата звернення: 18.03.2026).
3. Evergreens. Emotion AI: як працює розпізнавання емоцій. URL: https://evergreens.com.ua/ua/articles/emotion-ai.html (дата звернення: 18.03.2026).

УДК 004.732.2.056.5

ЗАСТОСУВАННЯ CISCO PACKET TRACER ДЛЯ СЕГМЕНТАЦІЇ ДОМАШНІХ WI-FI МЕРЕЖ

Муха В.С.
mukhavladstudakk@gmail.com
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м. Черкаси, Україна

У сучасних умовах стрімкого розвитку інформаційних технологій значно зростає кількість пристроїв, які підключаються до домашніх бездротових мереж. До них належать персональні комп’ютери, смартфони, планшети, телевізори, системи «розумного дому» та інші IoT-пристрої. Збільшення кількості підключених пристроїв призводить до зростання навантаження на мережу, що може негативно впливати на швидкість передачі даних, стабільність з’єднання та рівень інформаційної безпеки.

Одним із ефективних способів підвищення продуктивності та безпеки домашньої мережі є її оптимізація шляхом використання технології VLAN (Virtual Local Area Network). VLAN дозволяє логічно розділити одну фізичну мережу на декілька окремих сегментів. Завдяки цьому різні групи пристроїв можуть працювати в ізольованих мережевих середовищах, навіть якщо вони підключені до одного маршрутизатора або комутатора.

Використання VLAN у домашніх мережах дозволяє розділити пристрої за їх функціональним призначенням. Наприклад, можна створити окремий сегмент

для персональних пристроїв користувача, інший – для IoT-пристроїв, а також окрему мережу для гостьових підключень. Такий підхід зменшує ризик несанкціонованого доступу до внутрішніх ресурсів мережі та підвищує рівень захисту даних.

Важливим елементом оптимізації бездротової мережі є організація гостьового доступу. Гостьова Wi-Fi мережа дозволяє стороннім користувачам отримувати доступ до Інтернету без можливості підключення до внутрішніх ресурсів основної мережі. Це забезпечує додатковий рівень безпеки та дозволяє контролювати використання мережевих ресурсів.

Для моделювання та дослідження роботи домашньої мережі в межах даного проєкту найкраще підходить програмне середовище Cisco Packet Tracer. Це спеціалізований симулятор мереж, розроблений компанією Cisco Systems, який дозволяє створювати віртуальні комп'ютерні мережі, налаштовувати мережеве обладнання та аналізувати роботу різних мережевих технологій.

За допомогою Cisco Packet Tracer можна змоделювати кілька варіантів домашньої Wi-Fi мережі:

1. мережу без сегментації, де всі пристрої знаходяться в одному мережевому сегменті;
2. мережу з використанням VLAN, де пристрої поділені на окремі логічні сегменти;
3. мережу з гостьовим доступом, яка дозволяє стороннім користувачам підключатися до Інтернету без доступу до внутрішніх ресурсів мережі.

Для кожного сценарію можна використати відповідні налаштування мережевого обладнання, зокрема конфігурація VLAN, призначення портів комутатора та налаштування різних бездротових мереж .

Використання Cisco Packet Tracer дозволяє наочно дослідити роботу мережі, перевірити правильність конфігурації обладнання та порівняти ефективність роботи мережі в умовах наявності та відсутності сегментації. Це дає можливість оцінити переваги використання технології VLAN та гостьових

мереж для підвищення безпеки й продуктивності домашньої Wi-Fi інфраструктури.

Списки використаних джерел:

1. Tanenbaum A. S. Computer Networks: Pearson New International Edition. Pearson Education, Limited, 2013. 816 с.
2. Dell Networking SONiC: Як створити VLAN та призначити його Trunk та Access Switchport | Dell Ukraine. *Computer, Monitore und Technologielösungen | Dell Deutschland*. URL: https://www.dell.com/support/kbdoc/uk-ua/000217901/dell-networking-sonic-як-створити-vlan-та-призначити-його-trunk-та-access-switchport (дата звернення: 11.03.2026).
3. Kurose J., Ross K. Pearson EText Computer Networking: A Top-Down Approach -- Instant Access [Global Edition]. Pearson Education, Limited, 2021. 800 с.
4. Resource Hub: Get Packet Tracer, Virtual Machines, and More. *Cisco Networking Academy: Learn Cybersecurity, Python & More*. URL: https://www.netacad.com/resources/lab-downloads?courseLang=en-US (дата звернення: 11.03.2026).
5. Contributors to Wikimedia projects. Packet Tracer - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Packet_Tracer (дата звернення: 11.03.2026).

УДК 004.8

ВИКОРИСТАННЯ 3D МОДЕЛЮВАННЯ В GAME DEV

Шелестюк Є.Р.
shelestuk.ev@gmail.com
Черкаський державний фаховий бізнес-коледж
Подорошко Д.І.
м. Черкаси, Україна

3D моделювання в Game Dev займає важливе місце, оскільки саме через нього створюється візуальна частина гри. Будь-який об’єкт, який бачить гравець – персонаж, зброя, транспорт або навіть дрібні деталі оточення – існує у вигляді тривимірної моделі. Це дозволяє сформувати цілісний ігровий світ, який виглядає переконливо та живо. Сучасні ігри висувають високі вимоги до графіки, тому роль 3D моделювання постійно зростає.

Основою будь-якої моделі є полігональна сітка. Вона складається з вершин, ребер і граней, які формують геометрію об’єкта. Чим більше полігонів, тим вищий рівень деталізації, але водночас збільшується навантаження на систему. У результаті виникає необхідність знаходити баланс між якістю та продуктивністю. Саме через це в Game Dev активно застосовуються методи оптимізації.

Процес створення 3D моделі зазвичай починається з блокінгу. На цьому етапі формується загальний силует і визначаються пропорції об’єкта. Далі відбувається деталізація, де додаються дрібні елементи, що формують зовнішній вигляд. Після цього використовується текстурування, яке надає моделі кольору, фактури та реалістичності. Це дозволяє зробити об’єкт візуально складнішим без значного збільшення геометрії.

Важливу роль відіграють додаткові техніки, такі як нормал-мапи та карти висот. Вони створюють ілюзію дрібних деталей на поверхні моделі. У результаті навіть відносно проста геометрія може виглядати складною та деталізованою. Це особливо важливо для ігор, де необхідно зберігати високу продуктивність.

Для створення 3D моделей використовується спеціалізоване програмне забезпечення. Одним із найпоширеніших інструментів є Blender. Він є

безкоштовним і має широкий функціонал, що включає моделювання, текстурування та анімацію. Також активно використовується Autodesk Maya, яка добре підходить для складних проєктів і часто застосовується у великих студіях. Ще одним популярним інструментом є 3ds Max, який зручний для створення об’єктів середовища.

Крім основних програм, використовуються й додаткові інструменти. Наприклад, ZBrush застосовується для створення високодеталізованих моделей за допомогою цифрового скульптингу. Substance Painter дозволяє працювати з текстурами на більш глибокому рівні, додаючи ефекти зношення, бруду або металевих відблисків. Це дозволяє досягти більшої реалістичності об’єктів.

Таблиця 1 – Детальне порівняння інструментів для 3Д моделювання

№	Інструмент	Переваги	Недоліки
1	Blender	Безкоштовний і відкритий код; підтримка повного пайплайну (моделювання, анімація, рендер); велика спільнота та кількість аддонів	Незвичний інтерфейс для новачків; у великих студіях використовується рідше
2	Autodesk Maya	Потужні інструменти для анімації та ріггінгу; стандарт у великих студіях; гнучка інтеграція в пайплайн	Висока вартість; складний поріг входу
3	3ds Max	Зручний для створення середовища та архітектури; точне моделювання; хороша інтеграція з рендерами	Працює лише на Windows; слабші інструменти анімації
4	ZBrush	Надзвичайно деталізований скульптинг (десятки мільйонів полігонів); ідеальний для персонажів	Складний інтерфейс; не підходить як основний інструмент для всього пайплайну
5	Houdini	Процедурне моделювання; потужні симуляції (вибухи, фізика)	Дуже складний у вивченні; надлишковий для простих задач
6	Cinema 4D	Простий у використанні; сильний у motion-дизайні	Менше застосовується в геймдеві; платний

Після створення модель інтегрується в ігровий рушій. Найчастіше використовуються Unity або Unreal Engine. На цьому етапі налаштовуються матеріали, освітлення та фізичні властивості. У результаті модель стає частиною ігрового процесу та взаємодіє з іншими елементами сцени.

Таблиця 2 – Порівняння ігрових рушіїв

№	Рушій	Переваги	Недоліки
1	Unity	Кросплатформеність (понад 25 платформ); простий старт; підходить для мобільних і інді-проєктів	Менш потужна графіка порівняно з конкурентом; іноді потрібна додаткова оптимізація
2	Unreal Engine	Високоякісна графіка (Nanite, Lumen); зручний візуальний редактор; підходить для AAA-проєктів	Вищі вимоги до ПК; складніший для новачків
3	Godot	Безкоштовний і відкритий код; легкий та швидкий; зручний для інді-розробки	Менше можливостей для великої 3D графіки; слабша екосистема
4	CryEngine	Дуже реалістична графіка; потужний рендеринг	Складний у використанні; менша популярність і підтримка
5	Source (Valve)	Оптимізований і стабільний; добре підходить для FPS	Застарілі технології; обмежені можливості сучасної графіки

Не менш важливою є оптимізація. Для цього застосовуються різні підходи, наприклад створення кількох рівнів деталізації (LOD). Коли об’єкт знаходиться далеко від камери, використовується спрощена версія моделі. Це дозволяє зменшити навантаження на систему без помітної втрати якості.

3D моделювання в Game Dev поєднує технічні знання та творчий підхід. Саме завдяки цьому створюються ігрові світи, які виглядають переконливо та викликають інтерес у гравця. Внаслідок розвитку технологій цей напрям продовжує вдосконалюватися, відкриваючи нові можливості для створення ще більш реалістичних ігор.

Список використаних джерел:

1. Blender 5.1 Reference Manual. URL: https://docs.blender.org/manual/en/latest/?utm_medium=www-footer (дата звернення: 09.03.2026).
2. Autodesk Maya User Manual.URL: https://help.autodesk.com/view/MAYAUL/2026/ENU (дата звернення: 09.03.2026).

3. 3ds Max User Manual. URL: https://help.autodesk.com/view/3DSMAX/2025/ENU/ (дата звернення: 09.03.2026).
4. ZBrush Documentation. https://help.maxon.net/zbr/en-us/ (дата звернення: 09.03.2026).
5. Unity Documentation. https://docs.unity3d.com/Manual/index.html (дата звернення: 09.03.2026).
6. Unreal Engine 5.7 Documentation https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-5-7-documentation (дата звернення: 09.03.2026).

УДК 004.738.5:004.896

ІНТЕРНЕТ РЕЧЕЙ У ПОВСЯКДЕННОМУ ЖИТТІ: СИСТЕМА «РОЗУМНОГО БУДИНКУ»

Кудрявцева М. В.
ingamartynenko6@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В. Ю.
м. Черкаси, Україна

У сучасному цифровому суспільстві технології відіграють ключову роль у формуванні нових стандартів життя. Одним із найбільш перспективних напрямів є розвиток автоматизованих житлових систем, відомих як «розумний дім». Ця концепція не лише змінює уявлення про комфорт, але й сприяє більш раціональному використанню ресурсів, підвищенню рівня безпеки та оптимізації повсякденних процесів. В умовах сучасних викликів, зокрема енергетичних і соціальних, розумний дім стає не просто інновацією, а необхідністю.

Важливим аспектом функціонування розумного дому є централізована система управління, яка об’єднує всі пристрої в єдину мережу. Така система може працювати на основі спеціальних хабів або хмарних сервісів, що забезпечують доступ до керування з будь-якої точки світу. Користувач отримує

повний контроль над своїм житлом, включаючи можливість віддаленого моніторингу та налаштування параметрів роботи систем.

Особливу увагу варто приділити ролі датчиків у розумному домі. Саме вони збирають інформацію про стан середовища: температуру, вологість, освітленість, рух, якість повітря тощо. На основі цих даних система приймає рішення щодо подальших дій. Наприклад, при зниженні температури автоматично вмикається опалення, а при виявленні диму – сигналізація та система оповіщення.

Ще одним важливим напрямом є інтеграція розумного дому з мобільними технологіями. Сучасні смартфони стають універсальними пультами керування, які дозволяють контролювати всі аспекти житла. Крім того, розвиток голосових асистентів робить взаємодію з системою ще більш простою та інтуїтивною. Це особливо зручно для людей, які не мають технічного досвіду.

Варто також зазначити роль розумного дому в забезпеченні енергоефективності. Завдяки автоматичному регулюванню освітлення, опалення та роботи електроприладів можна значно зменшити споживання енергії. Це не лише знижує витрати, але й сприяє збереженню довкілля. У поєднанні з відновлюваними джерелами енергії розумний дім може стати майже автономною системою.

Значну роль відіграє і безпековий аспект. Окрім стандартних систем сигналізації, сучасні розумні будинки можуть використовувати біометричні технології, такі як розпізнавання обличчя або відбитків пальців. Це підвищує рівень захисту та зменшує ризик несанкціонованого доступу. Крім того, система може вести журнал подій, що дозволяє аналізувати всі дії в домі.

Економічний аспект розумного дому також заслуговує на увагу. Хоча початкові витрати можуть бути значними, у довгостроковій перспективі такі системи окупаються за рахунок економії ресурсів. Крім того, наявність розумних технологій підвищує ринкову вартість нерухомості, що робить її більш привабливою для покупців.

У контексті розвитку суспільства розумний дім є частиною ширшої концепції цифровізації. Він інтегрується з іншими системами, такими як «розумне місто», де всі елементи інфраструктури взаємодіють між собою. Це відкриває нові можливості для оптимізації міського життя, зменшення навантаження на ресурси та підвищення рівня комфорту громадян.

Не можна оминути і питання освіти та підготовки спеціалістів у цій сфері. Розвиток розумних технологій потребує кваліфікованих кадрів, здатних розробляти, встановлювати та обслуговувати такі системи. Тому важливо впроваджувати відповідні освітні програми та курси, які сприятимуть розвитку цієї галузі.

Отже, розумний дім є важливим елементом сучасного технологічного прогресу. Він поєднує інновації, комфорт, безпеку та економічну ефективність, створюючи новий стандарт житла. У майбутньому такі системи стануть ще більш доступними та функціональними, що дозволить їм зайняти провідне місце в житті людини.

Списки використаних джерел:

1. Історія розумного будинку-з чого все починалося? URL: https://www.smarthouse.ua/ua/istoriya-umnogo-doma-s-chego-vsenachinalos.html. (дата звернення: 11.03.2026).
2. Безпека передачі даних для інтернету речей. URL: https://www.csecurity.kubg.edu.ua/index.php/journal/article/view/107 (дата звернення: 11.03.2026).
3. IoT Privacy and Security Challenges for Smart Home Environments. URL: https://www.mdpi.com/2078-2489/7/3/44 (дата звернення: 11.03.2026).

УДК 004.7:004.9

VLAN ЯК ІНСТРУМЕНТ ЛОГІЧНОГО СЕГМЕНТУВАННЯ ТА БЕЗПЕКИ КОМП'ЮТЕРНИХ МЕРЕЖ

Шиян Д. С.
shyian811@gmail.com
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м. Черкаси, Україна

VLAN (віртуальна локальна мережа) є відповідним методом для логічного сегментування комп'ютерної мережі на різні області, незалежно від фактичного розташування підключених пристроїв. Суть цього рішення полягає в тому, що одна фізична мережа може бути розділена на кілька ізольованих логічних мереж, що може значно спростити управління, покращити безпеку та продуктивність мережі. VLAN дозволяє адміністратору групувати пристрої за функціями, відділами або типами послуг, навіть для тих, що підключені до різних комутаторів. Це забезпечує більшу гнучкість і легкість в управлінні мережею [1].

Найбільш поширені застосування VLAN – це корпоративні мережі підприємств. Великі організації мають кілька відділів: бухгалтерія, адміністративний, кадровий, IT-послуги, поділ на відділи. Якщо всі комп'ютери підключені до однієї мережі без поділу, це створює проблеми з безпекою і призводить до перевантаження мережевого трафіку. VLAN логічно розділяє мережу і надає відділам окремі середовища, в яких вони можуть працювати. Наприклад, трафік з одного відділу не впливає на інші підрозділи, а доступ до ресурсів може бути керований під наглядом мережевих адміністраторів.

Друге значне застосування VLAN – це великі локальні мережі з великою кількістю комп'ютерів та інших мережевих пристроїв. За відсутності логічного поділу в таких мережах буде багато широкомовних повідомлень, які отримуватимуть всі мережеві пристрої. В результаті мережа перевантажується, і швидкість її роботи знижується. VLAN дозволяє створити окремі домени широкомовлення, з яких надсилаються ці повідомлення. Таким чином, лише вузли в певному VLAN отримують трафік, що значно підвищує ефективність

мережі. Технологія VLAN також використовується в мережах, в яких одночасно передаються різні види трафіку, такі як дані, голос і відео.

У сучасних бізнесах IP-телефонія, відеоконференції та подібні послуги вимагають ефективної, швидкої та надійної форми транспортування в комунікаційній області. Це дозволяє VLAN розділяти трафік на різні частини. Одним з таких спеціалізованих сегментів є Voice VLAN для IP-телефонії, що дозволяє пріоритизувати деякі види даних і забезпечувати стабільну якість зв'язку навіть при перевантаженні даними [2].

Однією з важливих причин використання VLAN є підтримка безпеки мережі. Раціональне розділення компонентів мережі дозволяє групувати користувачів і пристрої в безпечний спосіб. Наприклад, сервери компанії або важливі ресурси можуть бути надані лише певному VLAN для конкретних користувачів. Ці ж ресурси не матимуть прямого доступу до інших кінцевих користувачів, що знаходяться в інших сегментах мережі. Це мінімізує можливість несанкціонованого доступу та витоку інформації. Крім того, сегмент мережі не поширюється на решту інфраструктури, коли виникають проблеми або атаки.

У випадку, коли мережа потребує гнучкості або має вміщувати нову конфігурацію, VLAN також є популярною технологією завдяки своїй практичності. У застарілих мережах зміна структури зазвичай вимагає перепідключення кабелів або переміщення пристроїв. У мережах з використанням VLAN більшість змін відбувається на рівні програмного забезпечення; налаштування комутаторів використовуються для зміни конфігураційних налаштувань. Наприклад, якщо комп'ютер користувача переміщується в інший відділ, єдина зміна, яка потрібна для цього користувача, - це переключення з одного VLAN на інший без зміни фізичного підключення. Це значно полегшує управління системою і зменшує час на обслуговування.

Централізоване управління мережею - це ще одна важлива концепція. VLAN дозволяють IT-персоналу призначати політики доступу, правила безпеки та налаштування з одного центрального місця по всій мережі. Це полегшує

управління великою кількістю машин швидше і зменшує кількість помилок у налаштуванні. Крім того, це критичний організаційний фактор для масштабування. Нові пристрої або підрозділи можуть швидко вводитися в відповідний VLAN, якщо вони заповнюють мережу. У навчальних закладах, центрах обробки даних та інших організаціях з великою кількістю користувачів впровадження VLAN також є ефективним. У таких мережах доступ зберігається в межах поділу між групами користувачів, наприклад, студентами, викладачами, адміністрацією тощо. VLAN дозволяє визначати та розділяти сегменти мережі для кожної групи, для більш послідовного доступу до ресурсів [3].

Таким чином, VLAN є ключовим інструментом для створення сучасних комп'ютерних мереж. Це корисно для логічного поділу мережі, високої безпеки, зменшення широкомовного трафіку, і спрощення адміністрування. Щодо корпоративних мереж, великих локальних мереж, систем з різними типами трафіку, а також середовищ, що вимагають гнучкості одночасно, VLAN рекомендується для використання конфігурації VLAN. Завдяки цим перевагам технологія VLAN широко впроваджується в сучасні мережеві інфраструктури і є одним з ключових елементів організації мережевих інфраструктур.

Список використаних джерел

1. Cyberset. VLAN: як працює і навіщо потрібен. URL: https://cyberset.com.ua/beginners/network/vlan-how-it-works-and-why-you-need-it/
2. Fiberroad. VLAN Explained: What Is VLAN and How Does It Work. URL: https://fiberroad.com/uk/resources/glossary/vlan-explained-what-is-vlan-how-does-it-work/
3. E-Server. Що таке VLAN: логіка технології і налаштування, реалізація VLAN в пристроях Cisco. URL: https://e-server.com.ua/uk/poradi/shho-take-vlan-logika-tehnologija-i-nalashtuvannja-realizacija-vlan-v-pristrojah-cisco

004.932.2.032.24

СИСТЕМА ЕЛЕКТРОННОГО ЗАПИСУ КЛІЄНТІВ У СФЕРІ ПЕРУКАРСЬКИХ ПОСЛУГ

Котляренко С.В.
kotletasb123@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В.Ю.
м. Черкаси, Україна

У сучасних умовах цифровізації побутового обслуговування автоматизація взаємодії з клієнтами є критично важливою для успішного функціонування бізнесу. Сфера перукарських послуг характеризується високою інтенсивністю записів, що потребує чіткого тайм-менеджменту та мінімізації помилок. Традиційні методи фіксації візитів поступово втрачають актуальність через ризик «накладок» у графіку та відсутність цілодобового доступу клієнта до послуги самозапису.

Проблема неефективного використання робочого часу майстрів у перукарнях часто пов'язана з людським фактором адміністратора. Застосування спеціалізованих систем електронного запису дозволяє не лише впорядкувати чергу, а й підвищити лояльність клієнтів за рахунок зручності бронювання 24/7. Крім того, автоматизація збору даних дає змогу власнику закладу отримувати об'єктивну аналітику щодо завантаженості персоналу та популярності окремих процедур.

На сьогодні існують як універсальні CRM-системи, так і вузькоспеціалізовані програмні продукти для салонів краси. Проте розробка власної системи або адаптація існуючих під конкретні потреби малого бізнесу залишається актуальною задачею. Основна перевага електронної системи перед паперовим журналом полягає у можливості автоматичного формування звітів, веденні бази даних клієнтів з історією їхніх уподобань та інтеграції з сервісами сповіщень.

У межах дослідження пропонується модель системи, що складається з таких модулів:

1. Модуль «Клієнт»: інтерактивна форма вибору послуги (стрижка, фарбування тощо), вибір майстра на основі його портфоліо та вільних годин у календарі.
2. Модуль «Адміністратор»: панель керування всіма записами, редагування розкладу майстрів, внесення перерв та вихідних днів, а також контроль фінансових надходжень.
3. Модуль «Аналітика та звіти»: автоматичне обчислення прибутку за день/місяць, визначення «пікових» годин завантаження та розрахунок заробітної плати майстрів.

Для забезпечення стабільної роботи системи доцільно використовувати клієнт-серверну архітектуру, із зберіганням інформації в базі даних. База даних повинна містити такі ключові сутності:

- Table_Clients: ПІБ, номер телефону, накопичувальна знижка.
- Table_Masters: ПІБ, спеціалізація, графік роботи, ставка оплати.
- Table_Services: назва послуги, тривалість у хвилинах, вартість матеріалів.
- Table_Bookings: унікальний ID запису, дата, час, посилання на клієнта та майстра.

Важливим етапом є інтеграція з API месенджерів (Viber, Telegram) або SMS-шлюзів. Це дозволяє реалізувати систему автоматичних нагадувань клієнту за 2-3 години до візиту, що значно знижує відсоток неявок.

Оскільки система передбачає збір та обробку персональних даних, необхідно забезпечити високий рівень безпеки. Це включає використання протоколів шифрування даних (SSL/TLS), розмежування прав доступу (майстер не має бачити фінансову звітність всього закладу) та регулярне створення резервних копій бази даних для запобігання втраті інформації.

Впровадження системи електронного запису є стратегічно важливим кроком для розвитку перукарського бізнесу. Це дозволяє оптимізувати щільність записів, виключити простій майстрів та забезпечити сучасний рівень сервісу. Використання такої системи трансформує хаотичний процес запису у

впорядкований цифровий актив, що сприяє масштабуванню бізнесу та підвищенню його прибутковості.

Список використаних джерел:

1. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ, 2016. 17 с.
2. Ткаченко П. О. Автоматизація сервісних підприємств: навч. посіб. Одеса: Астропринт, 2023. 180 с.
3. Програмне забезпечення для CRM-систем / за ред. М. В. Сидоренко. Київ: Техніка, 2024. 215 с.

УДК 004.8

ВИКОРИСТАННЯ THREE.JS ДЛЯ РОЗРОБКИ ВЕБ-ОРІЄНТОВАНИХ 3D-ДОДАТКІВ

Пустовіт М. В.
zxcmakson67@gmail.com
Черкаський державний фаховий бізнес-коледж
Дмитрюк В. В.
м. Черкаси, Україна

У сучасному цифровому середовищі активно розвиваються технології візуалізації даних та інтерактивної графіки. Одним із ключових інструментів для створення 3D-графіки у веббраузері є бібліотека Three.js, яка базується на технології WebGL і дозволяє реалізовувати складні тривимірні сцени без необхідності використання низькорівневого програмування графічних API [2].

Three.js є JavaScript-бібліотекою, яка значно спрощує роботу з 3D-графікою, надаючи високорівневі абстракції для створення сцен, камер, освітлення, матеріалів і анімацій. Це дозволяє розробникам швидко створювати інтерактивні веб-додатки, ігри, симулятори та навчальні системи. На рисунку 1 зображено схему поля зору камери (FOV) у Three.js.

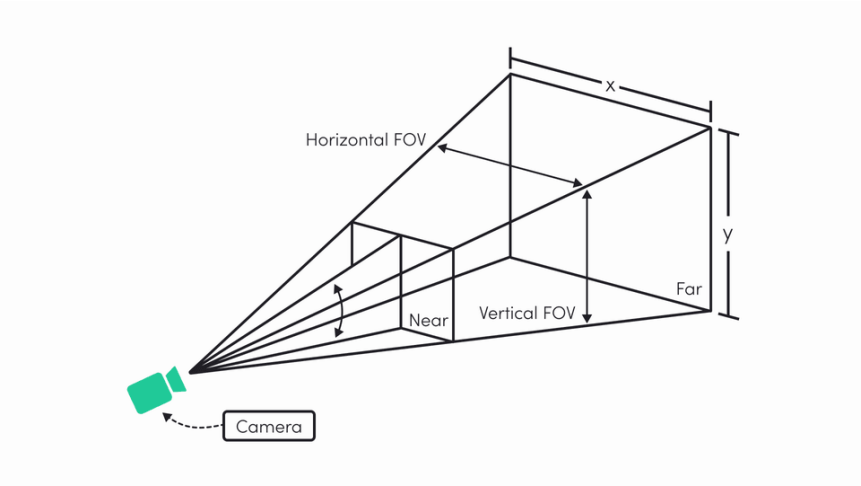


Рисунок 1 – Схема поля зору камери (FOV) у Three.js

Основними елементами Three.js є сцена (Scene), камера (Camera) та рендерер (Renderer). Сцена виступає контейнером для всіх об’єктів, камера визначає точку огляду, а рендерер відповідає за відображення графіки у браузері. Взаємодія між цими компонентами дозволяє створювати повноцінні тривимірні середовища [1].

Для побудови об’єктів у Three.js використовуються геометрії (Geometry) та матеріали (Material), які разом формують тривимірні моделі. Освітлення відіграє важливу роль у формуванні реалістичності сцени, оскільки визначає, як об’єкти відображають світло та тіні [3].

Three.js підтримує різні типи освітлення, зокрема [5]:

- Ambient Light – загальне освітлення сцени;
- Directional Light – напрямлене світло;
- Point Light – точкове джерело світла;
- Spot Light – прожектор.

Таблиця 1 – Основні компоненти Three.js

№	Компонент	Призначення	Переваги	Недоліки
1	Scene	Контейнер об’єктів	Простота організації	Вимагає оптимізації
2	Camera	Визначає точку огляду	Гнучкість	Потребує налаштування
3	Renderer	Відображення графіки	Висока продуктивність	Залежність від GPU
4	Geometry	Форма об’єктів	Великий вибір	Складність для великих моделей
5	Material	Зовнішній вигляд	Реалістичність	Впливає на продуктивність

Важливим аспектом використання Three.js є організація циклу рендерингу. Зазвичай застосовується функція анімації (render loop), яка безперервно оновлює сцену та відображає зміни. Це дозволяє реалізовувати рух об’єктів, взаємодію користувача та динамічні ефекти.

Крім того, Three.js підтримує обробку подій користувача, таких як рух миші, натискання клавіш та кліки. Це дає змогу створювати інтерактивні системи, включаючи 3D-ігри, тренажери та віртуальні середовища [4].

Однією з ключових переваг Three.js є можливість роботи в режимі реального часу. Завдяки використанню WebGL, обчислення виконуються на графічному процесорі (GPU), що забезпечує високу продуктивність навіть для складних сцен [2].

Three.js також широко застосовується у таких сферах:

- розробка веб-ігор;
- візуалізація даних;
- архітектурне моделювання;
- навчальні симулятори;
- доповнена та віртуальна реальність.

Однак використання Three.js має і певні обмеження. До них належать складність оптимізації великих сцен, залежність від продуктивності пристрою користувача та необхідність знання основ 3D-графіки [3].

Three.js є потужним інструментом для створення сучасних веб-орієнтованих 3D-додатків. Його використання дозволяє значно спростити процес розробки інтерактивної графіки, забезпечити високу продуктивність та створювати гнучкі програмні рішення. Завдяки своїм можливостям Three.js відкриває широкі перспективи для розвитку вебтехнологій та інтеграції тривимірної графіки у різні сфери діяльності.

Таким чином, застосування Three.js сприяє розвитку сучасних інформаційних систем, підвищує рівень взаємодії користувача з програмними продуктами та дозволяє створювати інноваційні цифрові рішення.

Список використаних джерел:

1. Cabello R. Three.js Documentation. URL: https://threejs.org/docs/ (дата звернення: 16.03.2026).
2. Khronos Group. WebGL Specification. URL: https://www.khronos.org/webgl/ (дата звернення: 16.03.2026).
3. Dirksen J. Learning Three.js. Packt Publishing, 2018.
4. Mozilla Developer Network. WebGL API. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API (дата звернення: 16.03.2026).
5. Stemkoski L. Three.js Examples. URL: http://stemkoski.github.io/Three.js/ (дата звернення: 16.03.2026).

УДК 004.94:620.92

ОПТИМІЗАЦІЯ РОЗМІЩЕННЯ СОНЯЧНИХ ПАНЕЛЕЙ НА БУДІВЛЯХ ЗА ДОПОМОГОЮ ПРОГРАМНИХ ЗАСОБІВ

Сивак Н. К.
ya.nazar256@gmail.com
Черкаський державний фаховий бізнес-коледж
Хотунов В. І.
м. Черкаси, Україна

Світова енергетична криза та прагнення до децентралізації електропостачання актуалізують питання ефективного використання відновлюваних джерел енергії в міських умовах. Україна, яка має значний потенціал сонячної енергії (середньорічна інсоляція становить 1200-1400 кВт·год/м²), поступово інтегрує фотоелектричні (ФЕ) панелі в архітектуру будівель. Особливої актуальності це набуває в умовах руйнування централізованої енергетичної інфраструктури внаслідок воєнних дій та необхідності забезпечення енергетичної автономії окремих будівель.

Одним із найбільш економічно доцільних напрямів використання сонячної енергії в міських умовах є інтеграція ФЕ-панелей на вже існуючі будівлі. Такий підхід не потребує додаткового відведення земельних ділянок, мінімізує витрати на монтажні конструкції та дозволяє наблизити генеруючі потужності до споживача. Однак ефективність такого рішення критично залежить від правильності вибору місць розташування панелей, їх орієнт ації та кута нахилу. За оцінками дослідників, помилки при виборі місць розміщення панелей можуть знижувати реальний виробіток електроенергії на 30-40% порівняно з потенційно можливим [1].

Традиційні методи проєктування сонячних електростанцій використовують усереднені показники інсоляції та спрощені геометричні моделі. Вони не враховують складну просторову конфігурацію сучасних будівель, наявність архітектурних елементів (парапетів, вентиляційних шахт, антен), що створюють затінення, а також взаємний вплив сусідніх будинків та зелених насаджень. Вирішення цієї проблеми потребує розробки

спеціалізованих програмних засобів для автоматизованої оптимізації просторового розташування сонячних модулів.

У даній роботі пропонується програмна реалізація оптимізаційної моделі, що поєднує геопросторовий аналіз із еволюційними обчисленнями. На першому етапі на основі цифрової моделі поверхні будівлі, яка формується з використанням відкритих геоданих OpenStreetMap та супутникових знімків Sentinel-2, будується інсоляційна карта. Для кожної елементарної ділянки площею 0,5×0,5 м розраховується річний потенціал сонячної енергії [2].

Другий етап передбачає розв'язання задачі оптимізації за допомогою генетичного алгоритму. Хромосома кодує набір позицій для розміщення N панелей, а фітнес-функція максимізує сумарний виробіток з урахуванням втрат через взаємне затінення. Генетичний алгоритм реалізує наступні оператори: селекція турнірним методом, одноточкове схрещування з ймовірністю 0,85 та мутація з ймовірністю 0,05.

Програмну реалізацію виконано мовою Python із використанням бібліотек NumPy для обчислень, Shapely для роботи з геометричними об'єктами, Rasterio для обробки растрових даних інсоляції та DEAP для реалізації генетичного алгоритму.

Для верифікації ефективності запропонованого методу було проведено порівняльний експеримент. Критерієм оцінки обрано річний виробіток електроенергії (кВт·год/рік) на одиницю встановленої потужності (1 кВт пікової потужності панелей).

Як видно з даних табл. 1, запропонований метод оптимізації на основі генетичного алгоритму забезпечує приріст річного виробітку електроенергії на 24,8-33,0% порівняно з рівномірним розміщенням та на 6,5-10,5% порівняно з жадібним алгоритмом. Найбільший ефект (33,0%) досягнуто для будівлі школи зі складною вальмовою конфігурацією даху, де вплив взаємного затінення є найбільш критичним. Найменший приріст (24,8%) зафіксовано для торгового центру з пласким дахом, що пояснюється відсутністю суттєвих природних перешкод для сонячного випромінювання.

Таблиця 1 – Результати порівняльного експерименту з оптимізації розміщення ФЕ-панелей

№	Тип будівлі	Конфігурація даху	Рівномірне розміщення, кВт·год/кВт	Жадібний алгоритм, кВт·год/кВт	Генетичний алгоритм, кВт·год/кВт	Приріст %
1	Житловий будинок	Двосхилий (аз. 45°/135°)	985	1180	1275	29,4
2	Школа	Вальмовий (складний)	910	1095	1210	33,0
3	Торговий центр	Плаский	1050	1240	1310	24,8
4	Адміністративна будівля	Односхилий (аз. 180°)	1020	1215	1290	26,5

Джерело: Розробка автора

Час обчислень для одного об’єкта (розмірність задачі – до 150 панелей) не перевищував 8 хвилин на стандартному ноутбуці (Intel Core i5, 16 ГБ RAM), що підтверджує придатність розробленого засобу для використання в реальній інженерній практиці. Статистична значущість отриманих результатів підтверджена за допомогою t-критерію Стьюдента ($p < 0,05$) [3].

Таким чином, розробка та дослідження програмного модуля для автоматизованого розміщення фотоелектричних панелей на поверхнях будівель є ефективним інструментом підвищення енергоефективності міської забудови. Запропонований підхід, що поєднує геопросторовий аналіз та генетичні алгоритми, дозволяє автоматизувати процес проєктування та зменшити капітальні витрати на одиницю встановленої потужності. Подальші дослідження будуть спрямовані на інтеграцію даного підходу, у технології цифрових двійників будівель, з обов’язковим врахуванням динаміки змін цін в електро енергії.

Список використаних джерел:

1. Голуб В.О., Савченко О.В. Моделювання інсоляції будівель для задач сонячної енергетики. Відновлювана енергетика. 2023. № 2(71). С. 24-31.
2. Ковальчук С.П., Бондаренко І.М. Геоінформаційні системи в енергетичному плануванні міст. Київ: Логос, 2024. 288 с.
3. Петренко О.А. Генетичні алгоритми в задачах оптимізації просторового розміщення об'єктів. Системні технології. 2024. № 5(148). С. 42-49. (дата звернення: 06.04.2026).

УДК 004.946

РОЗВИТОК КІБЕРСПОРТУ: ВІД ІНФРАСТРУКТУРИ ДО СОЦІАЛЬНОЇ ІНТЕГРАЦІЇ

Йовченко Н. В.
silverletter.y.n.v@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

У сучасному цифровому суспільстві кіберспорт перетворився з розважального явища на повноцінну індустрію, що поєднує інформаційні технології, медіа та спортивну діяльність. Активний розвиток інфраструктури кіберспорту, зокрема професійних ліг, стримінгових платформ і спеціалізованих арен, сприяє його глобалізації та популяризації серед молоді. Водночас кіберспорт відіграє важливу роль у соціальній інтеграції, створюючи нові можливості для комунікації, самореалізації та професійного розвитку.

Крім того, кіберспорт супроводжується як новими можливостями (працевлаштування, розвиток цифрових навичок), так і потенційними ризиками (ігрова залежність, нерівномірний доступ до ресурсів), що зумовлює необхідність їх комплексного наукового аналізу.

Кіберспорт у сучасному розумінні є організованою змагальною діяльністю у сфері відеоігор, що функціонує за принципами традиційного спорту та формує окрему індустрію з розвиненою інфраструктурою. За останні десятиліття

кіберспорт трансформувався з аматорських змагань у глобальний ринок із мільярдними обсягами фінансування та багатомільйонною аудиторією [1].

Історичний розвиток кіберспорту пов'язаний із технологічним прогресом у сфері обчислювальної техніки та мережевих технологій. Поява багатокористувацьких ігор, розвиток 3D-графіки та інтернет-інфраструктури стали ключовими чинниками формування сучасного кіберспортивного середовища [2]. Водночас, поширене у популярних джерелах спрощене трактування історії кіберспорту не враховує складних соціально-економічних процесів, що супроводжували його становлення.

Значну роль у розвитку кіберспорту відіграє інфраструктура, яка включає кіберспортивні арени, навчальні центри, онлайн-платформи та комп'ютерні клуби. В Україні спостерігається активне зростання цієї інфраструктури, що свідчить про інтеграцію країни у глобальний кіберспортивний простір [3]. Проте слід критично зазначити, що рівень державної підтримки та системність розвитку галузі залишаються нерівномірними.

Економічний аспект кіберспорту характеризується високою залежністю від спонсорства, рекламних контрактів і медіаправ. За оцінками аналітичних агентств, глобальний ринок кіберспорту перевищує 1 млрд доларів США, однак така оцінка потребує уточнення, оскільки включає суміжні індустрії (стримінг, рекламу, геймдев) [4]. Таким чином, реальний економічний вплив кіберспорту може бути як переоціненим, так і недооціненим залежно від методології підрахунків.

Соціальний вимір кіберспорту також є предметом наукової дискусії. З одного боку, кіберспорт сприяє розвитку цифрових компетентностей, комунікативних навичок та створює нові можливості для працевлаштування. З іншого – існують ризики надмірної залученості, залежності від ігор та негативного впливу на фізичне здоров'я [5]. Особливої уваги потребує питання використання кіберспорту як інструменту соціальної інтеграції вразливих груп населення, що потребує подальших емпіричних досліджень.

Критичний аналіз показує, що популяризація кіберспорту часто супроводжується перебільшенням його позитивного впливу та ігноруванням потенційних ризиків. Наприклад, твердження про гарантовані високі доходи кіберспортсменів не відображає реальної ситуації, оскільки лише незначний відсоток гравців досягає професійного рівня [4].

Перспективи розвитку кіберспорту пов’язані з інтеграцією новітніх технологій, зокрема віртуальної (VR) та доповненої реальності (AR), що можуть суттєво змінити формат змагань та взаємодії гравців. Водночас, подальший розвиток галузі залежатиме від регуляторної політики, інвестицій та рівня підготовки кадрів.

Кіберспорт є динамічною галуззю, що поєднує технологічні, економічні та соціальні аспекти. Його розвиток супроводжується як значними можливостями, так і певними ризиками. Ефективне використання потенціалу кіберспорту потребує комплексного підходу, що враховує інфраструктурні, соціальні та безпекові фактори.

Список використаних джерел:

1. Newzoo. Global Esports & Live Streaming Market Report. 2024. URL: https://newzoo.com (дата звернення: 15.03.2026).
2. Taylor T. L. Raising the Stakes: E-sports and the Professionalization of Computer Gaming. Cambridge: MIT Press, 2022. 328 p.
3. Вишняк П. Стан кіберспорту в Україні: інфраструктура, підтримка і роль держави. 2025. URL: https://vinnitsa.info (дата звернення: 15.03.2026).
4. Jenny S. E., Manning R. D., Keiper M. C., Olrich T. W. Virtual(ly) Athletes: Where eSports Fit Within the Definition of “Sport”. Quest. 2022. Vol. 74. P. 1–18.
5. Griffiths M. D. Adolescent Gaming and Gaming Disorder: A Review. Journal of Behavioral Addictions. 2021. Vol. 10. P. 1–12.

УДК 004.056

ОРГАНІЗАЦІЯ ЗАХИСТУ ВІД АТАК НА РІВНІ ПРОТОКОЛІВ IOT

Месевра О. О.
github853@gmail.com
Черкаський державний фаховий бізнес-коледж
Ратайчук П. Є.
м. Черкаси, Україна

Стрімкий розвиток та глобальне впровадження технологій Інтернету речей (IoT) у сучасні інфраструктурні рішення, починаючи від систем «розумного дому» і закінчуючи масштабними комплексами промислового Інтернету речей (IoT), вимагає розробки та впровадження принципово нових, надійних механізмів безпеки. Архітектура таких систем має свої специфічні особливості побудови: більшість кінцевих вузлів характеризуються вкрай обмеженими обчислювальними ресурсами, малим обсягом оперативної пам'яті та низьким рівнем енергоспоживання. Через ці апаратні обмеження використання традиційних, ресурсоємних мережевих протоколів та важких криптографічних алгоритмів стає неефективним або взагалі неможливим.

Саме тому основними стандартизованими протоколами прикладного рівня для взаємодії IoT-пристроїв на сьогодні є MQTT (Message Queuing Telemetry Transport) та CoAP (Constrained Application Protocol). Протокол MQTT функціонує на базі архітектури «публікація-підписка» (Publish-Subscribe) і використовує центральний вузол зв'язку – брокер, що дозволяє асинхронно обмінюватися повідомленнями між пристроями. Своєю чергою, протокол CoAP побудований за REST-архітектурою і працює поверх протоколу UDP, що мінімізує накладні витрати на встановлення з'єднання та робить його ідеальним для нестабільних мереж. Однак, незважаючи на їхню високу ефективність та оптимізованість для роботи в умовах обмежених ресурсів, базові специфікації цих протоколів часто не мають вбудованих суворих механізмів шифрування за замовчуванням, що робить їх вразливими до широкого спектра мережевих атак.

Актуальність даного дослідження зумовлена гострою необхідністю детального аналізу та класифікації векторів кібератак на рівні прикладних

протоколів IoT для подальшої розробки комплексних та дієвих систем захисту. Особливої уваги під час проєктування безпечних IoT-мереж потребують атаки типу відмови в обслуговуванні (DoS/DDoS), перехоплення та модифікації даних (Man-in-the-Middle), а також атаки, що базуються на повторній передачі повідомлень (Replay-атаки).

Теоретичний аналіз показує, що атака типу відмови в обслуговуванні (DoS) у середовищі MQTT найчастіше реалізується шляхом генерації зловмисником надлишкової кількості запитів на підключення до брокера (Connection Flooding) або масової публікації повідомлень у певні топіки. Це призводить до критичного вичерпання ресурсів брокера (завантаження процесора, переповнення оперативної пам'яті), внаслідок чого він втрачає здатність обробляти легітимні запити від авторизованих сенсорів чи контролерів. Виявити таку аномалію можливо за допомогою систем моніторингу мережі, які фіксують різке зростання кількості вхідних пакетів із певних IP-адрес, а також стрімке збільшення затримки відповіді сервера.

Не менш небезпечними є атаки типу Man-in-the-Middle (MITM). Оскільки базові версії MQTT та CoAP можуть передавати дані у відкритому вигляді (Plaintext), зловмисник, що знаходиться в одній мережі з пристроями, здатний перехоплювати телеметрію. MITM-атаки дозволяють зловмиснику не лише пасивно читати конфіденційні дані, а й активно модифікувати їх перед доставкою брокеру або клієнту, що може призвести до хибного спрацьовування виконавчих механізмів в IoT-системі.

Replay-атаки (повторна передача повідомлень) спрямовані на перехоплення валідних, вже сформованих пакетів управління та їхню подальшу багаторазову відправку в мережу. Зловмисник імітує дії авторизованого клієнта, що є критичною вразливістю для систем, де не реалізовано належне шифрування, не використовуються часові мітки (timestamps) або унікальні ідентифікатори сесій. Визначення та виявлення цих загроз передбачає глибокий інспекційний аналіз мережевого трафіку (Deep Packet Inspection), пошук дубльованих

послідовностей пакетів та моніторинг аномалій у часових інтервалах надходження повідомлень.

У рамках подальшого практичного дослідження планується розробка та розгортання віртуального тестового середовища для моделювання вищезазначених векторів атак на протоколи MQTT та CoAP. Головним завданням практичної частини стане дослідження та фіксація ключових мережевих і системних параметрів. Зокрема, будуть аналізуватися такі метрики: затримка передачі повідомлень (Latency), відсоток втрат мережевих пакетів (Packet Loss Rate), загальна пропускна здатність каналу (Throughput), а також рівень навантаження на обчислювальні ресурси мережевих вузлів (використання CPU та RAM) під час штатної роботи та в умовах активної кібератаки. Для детального аналізу структури перехоплених пакетів та виявлення вразливостей застосовуватимуться спеціалізовані засоби мережевого моніторингу та аналізатори трафіку, такі як Wireshark.

Додатково планується дослідити ефективність впливу криптографічних протоколів TLS для MQTT та DTLS для CoAP на загальну продуктивність мережі. Буде проведено порівняльний аналіз швидкодії системи до та після впровадження механізмів парольної аутентифікації та списків контролю доступу (ACL), які виступають базовими методами протидії несанкціонованому доступу. Комплексний аналіз цих параметрів дозволить сформувати обґрунтовані рекомендації щодо організації безпеки в IoT-системах, збалансувавши рівень захисту з допустимими втратами продуктивності.

Список використаних джерел:

1. Banks A., Gupta R. MQTT Version 5.0. *OASIS Standard*. 2019. URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html (дата звернення: 25.02.2026).
2. Shelby Z., Hartke K., Bormann C. The Constrained Application Protocol (CoAP). *Internet Engineering Task Force (IETF), RFC 7252*. 2014. URL: https://datatracker.ietf.org/doc/html/rfc7252 (дата звернення: 01.03.2026).

3. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3. *Internet Engineering Task Force (IETF), RFC 8446*. 2018. URL: https://datatracker.ietf.org/doc/html/rfc8446 (дата звернення: 02.03.2026).
4. Rescorla E., Tschofenig H., Modadugu N. Datagram Transport Layer Security Version 1.2. *Internet Engineering Task Force (IETF), RFC 6347*. 2012. URL: https://datatracker.ietf.org/doc/html/rfc6347 (дата звернення: 02.03.2026).
5. Eclipse Mosquitto: An open source MQTT broker. *Eclipse Foundation*. URL: https://mosquitto.org/ (дата звернення: 05.03.2026).
6. Docker Security. *Docker Documentation*. URL: https://docs.docker.com/engine/security/ (дата звернення: 06.03.2026).
7. Wireshark User's Guide. *Wireshark Foundation*. URL: https://www.wireshark.org/docs/wsug_html_chunked/ (дата звернення: 09.03.2026).

УДК 004.946

ВИКОРИСТАННЯ ДОПОВНЕНОЇ РЕАЛЬНОСТІ У МЕДИЦИНІ: ВІД ДІАГНОСТИКИ ДО ОПЕРАЦІЙ

Левандовський Д. О.
Levandovskijsenis2@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Доповнена реальність (AR) є сучасною технологією, що поєднує реальне середовище з віртуальними об'єктами шляхом накладання цифрової інформації (зображень, 3D-моделей, тексту) в режимі реального часу [1]. У медичній галузі застосування AR відкриває нові можливості для підвищення точності діагностики, удосконалення підготовки медичного персоналу та оптимізації хірургічних втручань [2].

З огляду на зростання вимог до якості медичних послуг і безпеки пацієнтів, впровадження інноваційних технологій, зокрема доповненої реальності, набуває особливої актуальності [3].

Основою функціонування технології доповненої реальності є використання камер, сенсорів, спеціалізованого програмного забезпечення та пристроїв відображення, таких як окуляри, планшети та смартфони [1]. Це дозволяє лікарю отримувати додаткову інформацію про пацієнта без відриву від процесу обстеження або операції, що сприяє підвищенню ефективності прийняття клінічних рішень і зниженню ймовірності помилок.

У сфері діагностики AR застосовується для візуалізації внутрішніх органів на основі даних комп'ютерної томографії, магнітно-резонансної томографії та ультразвукових досліджень. Технологія дозволяє накладати тривимірні моделі органів на тіло пацієнта, що покращує розуміння анатомічних особливостей і сприяє точнішій локалізації патологій, зокрема пухлин, судинних аномалій та складних травм [2].

Одним із ключових напрямків використання AR є хірургія. Під час оперативних втручань технологія забезпечує візуалізацію критично важливих анатомічних структур, таких як судини, нерви та новоутворення, що дозволяє підвищити точність маніпуляцій і зменшити їх інвазивність. Крім того, AR використовується на етапі передопераційного планування, що дає можливість моделювати хід операції та прогнозувати її результати [3].

Використання доповненої реальності у підготовці медичних працівників сприяє підвищенню якості освіти. Завдяки інтерактивним тривимірним моделям і симуляціям студенти мають можливість відпрацьовувати практичні навички у безпечному середовищі, що знижує ризики для пацієнтів і підвищує рівень професійної підготовки [2].

У сфері реабілітації AR застосовується для створення інтерактивних програм відновлення, що дозволяють пацієнтам виконувати вправи у віртуальному середовищі. Це підвищує мотивацію до лікування та ефективність відновлення після травм або неврологічних захворювань [3].

Попри значні переваги, використання AR у медицині супроводжується низкою викликів, серед яких висока вартість обладнання, потреба у

спеціалізованій підготовці персоналу та необхідність забезпечення захисту медичних даних [2].

Отже, доповнена реальність є перспективною технологією, що суттєво трансформує сучасну медицину, забезпечуючи підвищення точності діагностики, ефективності лікування та якості підготовки медичних фахівців. Подальший розвиток і впровадження AR сприятимуть удосконаленню медичних послуг та підвищенню рівня безпеки пацієнтів, попри наявні технічні й організаційні обмеження.

Список використаних джерел:

1. Azuma R. A Survey of Augmented Reality. URL: https://www.cs.unc.edu/~azuma/ARpresence.pdf (дата звернення: 23.03.2026).
2. Barsom E., Graafland M., Schijven M. Systematic review on the effectiveness of augmented reality applications in medical training. URL: https://pubmed.ncbi.nlm.nih.gov/28183672/ (дата звернення: 23.03.2026).
3. Microsoft HoloLens in Healthcare. URL: https://www.microsoft.com/en-us/hololens/industry-healthcare (дата звернення: 23.03.2026).

УДК 004.942:681.5

ЗАСТОСУВАННЯ ЦИФРОВИХ ДВІЙНИКІВ В АВТОМАТИЗОВАНИХ СИСТЕМАХ КЕРУВАННЯ

Самойлов О.О.
2005samoylow@gmail.com
Черкаський державний фаховий бізнес-коледж
Швиденко А.В.
м. Черкаси, Україна

У сучасних умовах стрімкого розвитку цифрових технологій та автоматизації виробництва важливу роль відіграє впровадження інноваційних підходів до управління складними технічними системами. Одним із таких підходів є використання цифрових двійників, які є ключовим елементом

концепції Індустрії 4.0. Цифровий двійник являє собою динамічну віртуальну модель фізичного об’єкта, процесу або системи, яка відображає їхній стан у реальному часі на основі даних, отриманих із сенсорів та інформаційних систем [1].

Основною метою застосування цифрових двійників є підвищення ефективності функціонування автоматизованих систем керування (АСК), а також забезпечення їх гнучкості, адаптивності та надійності. Завдяки використанню цифрових моделей стає можливим не лише відображення поточного стану системи, але й прогнозування її поведінки в майбутньому, що є критично важливим для складних виробничих процесів [1, 2].

Цифрові двійники активно інтегруються з сучасними технологіями, такими як Інтернет речей (IoT), великі дані (Big Data), штучний інтелект та машинне навчання. Використання сенсорних мереж дозволяє отримувати значні обсяги даних у режимі реального часу, які обробляються аналітичними системами та використовуються для постійного оновлення цифрової моделі [2].

Однією з ключових переваг цифрових двійників є можливість проведення віртуальних експериментів. Це дозволяє тестувати різні режими роботи системи, оптимізувати технологічні параметри та оцінювати ефективність впровадження нових рішень без ризику для реального виробництва. Такий підхід значно знижує витрати на модернізацію обладнання та підвищує якість управління процесами [3].

Особливе значення цифрові двійники мають у сфері прогнозного обслуговування. Аналізуючи історичні та поточні дані, система здатна виявляти ознаки потенційних несправностей та попереджати про необхідність технічного обслуговування. Це дозволяє мінімізувати простої обладнання та підвищити надійність виробничих систем [3, 4].

В автоматизованих системах керування цифрові двійники виконують широкий спектр функцій, серед яких моніторинг, діагностика, оптимізація та підтримка прийняття рішень. Вони дозволяють створювати кіберфізичні

системи, у яких фізичні та цифрові компоненти тісно взаємодіють між собою, забезпечуючи адаптивне управління виробництвом [1, 4].

Незважаючи на значні переваги, впровадження цифрових двійників супроводжується певними труднощами. До них належать висока вартість розробки та впровадження, складність інтеграції з існуючими інформаційними системами, а також необхідність забезпечення належного рівня кібербезпеки [2].

Перспективи розвитку технології цифрових двійників пов’язані з подальшим удосконаленням методів обробки даних, розвитком штучного інтелекту та розширенням можливостей обчислювальних систем. Очікується, що в майбутньому цифрові двійники стануть невід’ємною частиною більшості автоматизованих систем керування, забезпечуючи їх автономність та інтелектуальність [5].

Таким чином, застосування цифрових двійників в автоматизованих системах керування є перспективним напрямом розвитку сучасних технологій, що дозволяє значно підвищити ефективність, надійність та гнучкість виробничих процесів.

Список використаних джерел:

1. Tao F., Qi Q., Liu A., Nee A. Y. C. Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0: correlation and comparison // Engineering. – 2019. – Т. 5, № 4. – С. 653–661.
2. Hu W., Zhang T., Deng X., Liu Z., Tan J. Digital twin: a state-of-the-art review of its enabling technologies, applications and challenges // Journal of Intelligent Manufacturing and Special Equipment. – 2021. – Т. 2, № 1. – С. 1–34.
3. Qi Q., Tao F., Zhang H., Nee A. Y. C. Enabling technologies and tools for digital twin // Journal of Manufacturing Systems. – 2019. – Т. 58. – С. 3–21.
4. Kritzinger W., Karner M., Traar G., Henjes J., Sihn W. Digital Twin in manufacturing: A categorical literature review and classification // IFAC-PapersOnLine. – 2018. – Т. 51, № 11. – С. 1016–1022.

5. Grieves M., Vickers J. Digital Twin: Mitigating unpredictable, undesirable emergent behavior in complex systems // Transdisciplinary Perspectives on Complex Systems. – Cham: Springer, 2017. – С. 85–113.

УДК 004.056:621.396

ПРАКТИЧНЕ МОДЕЛЮВАННЯ ТА АНАЛІЗ ЕФЕКТИВНОСТІ 5G-МЕРЕЖ ДЛЯ СИСТЕМ ІНТЕРНЕТУ РЕЧЕЙ (IOT)

Поштовий Д.О.
postovijdanilo@gmail.com
Черкаський державний фаховий бізнес-коледж
Ратайчук П.Є.
м. Черкаси, Україна

Стрімкий розвиток концепції Інтернету речей (IoT) вимагає створення надійної, масштабованої та енергоефективної мережевої інфраструктури. Традиційні мобільні мережі не завжди здатні впоратися з величезною кількістю одночасних підключень та забезпечити мінімальну затримку передачі даних, яка є критичною для таких сфер, як Smart City, Smart Industry та автономний транспорт. Впровадження мереж п'ятого покоління (5G) вирішує ці проблеми завдяки використанню новітніх технологій, таких як Massive MIMO, Network Slicing (мережеві зрізи) та Edge Computing (периферійні обчислення)[1]. Особливу увагу варто приділити технологіям доступу, спеціально розробленим для IoT, зокрема NB-IoT та LTE-M, які забезпечують високу енергоефективність пристроїв.

Метою даного дослідження є практичне моделювання 5G-мережі для підтримки великої кількості IoT-пристроїв та оцінка її ефективності за основними показниками якості обслуговування (QoS)[2].

Для проведення експерименту було обрано програмне середовище симуляції мереж NS-3 із використанням модулів 5G NR. Програмна реалізація моделі здійснювалася мовою C++ з підключенням спеціалізованих модулів, таких як nr-module для симуляції мережі п'ятого покоління та internet-module. Розроблена топологія мережі включає базову станцію (gNodeB), ядро мережі 5G

(5GC), сервер обробки даних та від 50 до 500 IoT-вузлів (UE), розташованих на площі 500 × 500 метрів. Моделювання відбувалося у частотному діапазоні 3.5 GHz зі смугою пропускання 100 MHz. Основний тип генерованого трафіку – телеметрія та короткі пакети сенсорних даних (Payload 64–256 байт) з інтервалом передачі від 1 до 10 секунд[2].

Для формалізації показників ефективності роботи мережі були використані наступні математичні моделі:

1. Пропускна здатність мережі (Throughput), що визначається як відношення сумарного обсягу прийнятих даних до часу симуляції.
2. Середня затримка передачі (Delay_avg).
3. Коефіцієнт успішної доставки пакетів (Packet Delivery Ratio, PDR).
4. Загальне енергоспоживання IoT-пристрою (E_total), яке є критичним показником для автономних сенсорів і складається з енергії передачі, прийому та режиму очікування.

Під час дослідження було проведено серію експериментів із базовим навантаженням (50 пристроїв), масштабуванням кількості вузлів до 500, інтенсивним трафіком (інтервал передачі 0.5 с) та впровадженням Edge Computing. Збір та обробка статистичних даних виконувалися шляхом їх експорту у формат CSV із подальшим аналізом та візуалізацією за допомогою мови програмування Python (бібліотеки pandas та matplotlib). Це дозволило побудувати наочні графіки залежності пропускної здатності та затримки від кількості активних вузлів [2, 3].

Аналіз отриманих результатів показав високу масштабованість 5G-мережі та здатність стабільно обслуговувати сотні IoT-пристроїв. Було встановлено, що середня затримка передачі становить менше 10–20 мс. Використання технології Edge Computing (перенесення обчислень на край мережі) дозволило додатково знизити затримку на 20–30%. Водночас, експериментальним шляхом було визначено критичний поріг навантаження: при перевищенні кількості у 400–500 активних IoT-пристроїв на одну базову станцію пропускна здатність починає знижуватися, а затримка різко зростає.

Таким чином, результати моделювання підтверджують високу ефективність використання 5G-мереж для розгортання масштабних IoT-систем. Для подальшої оптимізації роботи та запобігання перевантаженням рекомендується застосовувати механізми балансування навантаження між базовими станціями, використовувати мережеві зрізи (Network Slicing) для пріоритезації критичного трафіку та розширювати застосування периферійних обчислень (Edge Computing) [4].

Список використаних джерел:

1. *Кафедра інфокомунікаційної інженерії імені В.В. Поповського*. URL: https://ice.nure.ua/wp-content/uploads/2024/01/57_Popovska-Ie.O.-Marchuk-V.S._Str.191-193.pdf (дата звернення: 08.03.2026).
2. ns-3 Manual – Manual. *ns-3 | a discrete-event network simulator for internet systems*.
   URL: https://www.nsnam.org/docs/release/3.35/manual/html/index.html (дата звернення: 08.03.2026).
3. Management and Orchestration Standards for 5G. *3GPP – The Mobile Broadband Standard*. URL: https://www.3gpp.org/technologies/sa5-management-5g-h11 (дата звернення: 10.03.2026).
4. *DSpace: ELAKPI: Репозитарій КПІ ім. Ігоря Сікорського*. URL: https://ela.kpi.ua/server/api/core/bitstreams/6b18d6b4-1f7c-471c-afef-0e6c244b3acc/content (дата звернення: 11.03.2026).

УДК 004.8

РОЗРОБКА ІНТЕРАКТИВНОГО ВЕБПОРТАЛУ ДЛЯ УПРАВЛІННЯ ПЕРСОНАЛЬНИМИ ПЛАНАМИ

Удод В.В.
29042018ul@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В.Ю.
м. Черкаси, Україна

Сучасний ритм життя вимагає від кожної людини високого рівня самоорганізації, проте стандартні методи планування часто виявляються малоефективними через відсутність гнучкості та інтерактивності. Створення спеціалізованого вебпорталу вбачається логічним кроком у вирішенні проблеми систематизації щоденних завдань, тренувань та харчування. В основу проєкту покладено ідею формування єдиного цифрового простору, де користувач зможе отримати не просто статичний список справ, а динамічну систему, що адаптується під конкретні потреби.

Під час проектування особлива увага приділяється майбутній архітектурі бази даних та інтерфейсу користувача. Визначено, що ключовим аспектом успішного вебпорталу має стати інтуїтивно зрозуміла навігація, яка не перевантажує зайвою інформацією. Використання сучасних фреймворків дозволить реалізувати швидкий відгук сторінок та забезпечити стабільну роботу на різних типах пристроїв. У процесі підготовки технічного завдання виокремлюються основні модулі: блок персоналізації, календарний планувальник та система моніторингу прогресу.

Впровадження інтерактивних елементів, таких як динамічні графіки досягнень та автоматичні сповіщення, дозволить значно підвищити рівень залученості. Аналіз існуючих аналогів на ринку виявляє дефіцит рішень, які б поєднували в собі функції фітнес-трекера та менеджера завдань без необхідності перемикання між різними додатками. Це дає підстави стверджувати, що інтеграція декількох напрямків життєдіяльності в одну платформу є найбільш перспективним шляхом розвитку персональних вебсервісів.

Окремим етапом планується робота над безпекою даних. Оскільки портал передбачатиме зберігання особистої інформації, доцільним є впровадження протоколів шифрування та багаторівневої системи автентифікації. Це дозволить мінімізувати ризики несанкціонованого доступу, що є критично важливим для сучасних вебпроєктів. Технічна реалізація базуватиметься на клієнт-серверній моделі, де фронтенд частина відповідатиме за візуалізацію, а бекенд забезпечуватиме логіку обробки запитів та взаємодію з базою даних.

На етапі майбутнього тестування передбачається перевірка стійкості системи до великих навантажень та коректності відображення контенту в різних браузерах. Можливі недоліки в логіці розподілу ресурсів планується усувати шляхом оптимізації програмного коду на ранніх стадіях розробки. Це має забезпечити високу швидкість завантаження сторінок навіть при низькій швидкості інтернет-з’єднання. Використання адаптивної верстки гарантуватиме коректну роботу порталу як на мобільних телефонах, так і на стаціонарних комп’ютерах.

Запропонований інструментарій дасть можливість автоматизувати рутинні процеси, звільняючи час для більш важливих справ. Завдяки впровадженню алгоритмів персоналізації, кожен відвідувач отримуватиме унікальний досвід взаємодії з платформою. Вказаний підхід перетворює портал з простого сховища даних на повноцінного асистента у досягненні поставлених цілей. Поєднання візуальної привабливості та функціональної потужності розглядається як фундамент для залучення та утримання цільової аудиторії.Внаслідок реалізації описаних рішень очікується отримання конкурентоспроможного продукту, що відповідає актуальним вимогам веб-розробки. Проектована система має продемонструвати високу ефективність у питаннях менеджменту часу та особистих ресурсів. Подальший розвиток ідеї може бути пов’язаний із впровадженням елементів штучного інтелекту для ще більш точного прогнозування та коригування планів користувачів.

Список використаних джерел:

1. Google Cloud Security Best Practices. URL: https://cloud.google.com/security/best-practices (дата звернення: 19.03.2026).
2. React Documentation. URL: https://react.dev/ (дата звернення: 19.03.2026).
3. OWASP Top Ten Project. URL: https://owasp.org/www-project-top-ten/ (дата звернення: 19.03.2026).
4. Web Vitals: Metrics for a healthy site. URL: https://web.dev/vitals/ (дата звернення: 19.03.2026).

УДК 004.932.2.032.24

ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ ЗАДАЧ РОЗПІЗНАВАННЯ ЛЮДИНИ У РЕАЛЬНОМУ ЧАСІ

Шелег А.Р.
zei877x.34@gmail.com
Черкаський державний фаховий бізнес-коледж
Злочевська–Краснощок Д.С.
м. Черкаси, Україна

Розвиток систем інтелектуального відеоспостереження, безпілотних літальних апаратів та робототехніки поставив перед фахівцями з комп’ютерної інженерії складне завдання автоматизації розпізнавання об’єктів у відеопотоці. Однією з найбільш пріоритетних задач у цьому напрямку є детекція людини, що вимагає не лише високої точності класифікації, а й здатності працювати в режимі реального часу (Real-time detection). Сучасні алгоритми комп'ютерного зору пройшли шлях від класичних методів аналізу градієнтів до глибоких нейронних мереж, проте вибір конкретної архітектури завжди залишається компромісом між швидкістю обробки кадрів (FPS) та середньою точністю (mAP) [1]. Актуальність даного дослідження зумовлена необхідністю систематизації сучасних підходів для їх ефективного впровадження на апаратно обмежених платформах.

Історично першим ефективним методом розпізнавання людей був алгоритм на основі гістограм орієнтованих градієнтів (HOG) у поєднанні з

методом опорних векторів (SVM). Цей підхід базується на виділенні характерних контурів людської фігури та їх статистичній класифікації. Хоча HOG демонстрував непогані результати на статичних зображеннях, він виявився недостатньо ефективним для складних сцен із мінливим освітленням та перекриттям об’єктів[4]. З появою глибокого навчання (Deep Learning) фокус досліджень змістився до використання згорткових нейронних мереж (CNN), які здатні самостійно виділяти ієрархічні ознаки об’єктів, значно випереджаючи класичні методи за всіма показниками якості.

Сучасні нейромережеві детектори поділяються на два основні типи: двостадійні та одностадійні. Двостадійні алгоритми, яскравим представником яких є архітектура Faster R-CNN, спочатку генерують регіони з потенційними об’єктами, а потім виконують їх класифікацію та уточнення координат. Такий підхід забезпечує найвищу точність детекції, проте через високу обчислювальну складність він часто не здатний забезпечити стабільну роботу в реальному часі на побутових відеокартах або мобільних процесорах [7]. Саме тому для задач відеоаналітики частіше обирають одностадійні детектори, які виконують регресію координат об'єктів та їх класифікацію за один прохід мережі.

Найпопулярнішим сімейством одностадійних алгоритмів є YOLO (You Only Look Once). За останні роки це сімейство еволюціонувало від базових версій до надпотужних YOLOv8 та YOLOv10. Головною особливістю архітектури YOLO є розбиття зображення на сітку, де кожен осередок відповідає за прогнозування наявності об’єкта та його меж [2]. Алгоритм демонструє вражаючу швидкість, що дозволяє обробляти відеопотік з частотою понад 60-100 кадрів на секунду, що є критичним для систем безпеки. Конкурентним рішенням є архітектура SSD (Single Shot MultiBox Detector), яка використовує набір фільтрів різного масштабу для детекції об’єктів різного розміру, що дозволяє їй бути стабільнішою при виявленні дрібних фігур людей на великій відстані [3].

Аналіз апаратних аспектів впровадження показує, що для мобільних систем та вбудованих пристроїв, які часто використовуються в комп'ютерній інженерії, доцільно застосовувати полегшені версії алгоритмів, такі як

MobileNet-SSD або YOLO-tiny. Ці архітектури використовують глибинно-роздільні згортки (depthwise separable convolutions), що дозволяє зменшити кількість параметрів мережі в десятки разів при збереженні прийнятної точності [5]. Крім того, важливу роль відіграє програмна оптимізація через фреймворки на кшталт TensorRT або OpenVINO, які дозволяють максимально використовувати потужність конкретного заліза (GPU чи NPU) [6].

Отже, порівняльний аналіз демонструє, що для задач детекції людини у реальному часі оптимальним вибором є алгоритми сімейства YOLO завдяки їхній високій продуктивності. Однак, якщо пріоритетом є детекція малих об’єктів у складних умовах, архітектура SSD може показати кращу стабільність. Майбутнє розвитку галузі лежить у площині використання Vision Transformers (ViT), які обіцяють ще вищу точність, проте наразі вимагають значних ресурсів для обробки відео в реальному часі. Вибір конкретного рішення повинен ґрунтуватися на ретельному аналізі цільового обладнання та вимог до швидкодії конкретної системи [4].

Список використаних джерел:

1. Антонюк А. С., Ваврук П. М. Методи та засоби інтеграції штучного інтелекту в системи Інтернету речей. Науковий вісник НЛТУ України. 2021. Т. 31. № 2. С. 105–110.
2. Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. P. 7263–7271.
3. Liu W., et al. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision. Springer, Cham, 2016. P. 21–37.
4. Борисенко В. В. Порівняльний аналіз сучасних методів детектування об’єктів на зображеннях. Технічні науки та технології. 2022. № 2(28). С. 17–25.
5. Howard A. G., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861. 2017. 15 p.

6. Sarker I. H. Computer Vision AI: Techniques, Applications and Real-World Case Studies. SN Computer Science. 2023. Vol. 4. No. 1. P. 1–18.
7. Ren S., et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. Vol. 39. No. 6. P. 1137–1149.

УДК 004.925.8:523.2

ВИКОРИСТАННЯ БІБЛІОТЕКИ THREE.JS ДЛЯ СТВОРЕННЯ ІНТЕРАКТИВНИХ МОДЕЛЕЙ НЕБЕСНИХ ТІЛ У ВЕБ СЕРЕДОВИЩІ

Овчаренко Д.В.
diana.ovcharenko1331@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В.Ю.
м. Черкаси, Україна

Стрімкий розвиток веб-технологій відкриває широкі можливості для створення складних графічних рішень, які раніше були доступні лише у спеціалізованому програмному забезпеченні. Дослідження фокусується на практичних аспектах розробки інтерактивних 3D-моделей планет та інших космічних об'єктів за допомогою бібліотеки Three.js. Цей інструмент є потужною надбудовою над WebGL, що дозволяє працювати з тривимірним простором безпосередньо у браузері без необхідності встановлення додаткових плагінів.

Вибір Three.js як основного інструментарію зумовлений здатністю бібліотеки значно спрощувати роботу з 3D-сценами, об’єктами та освітленням. У межах дослідження було детально розглянуто ключові етапи побудови віртуального простору: ініціалізація сцени, конфігурація перспективної камери та налаштування візуалізатора. Саме така структура забезпечує фундамент для подальшого наповнення порталу контентом.

Для досягнення високого рівня реалістичності небесних тіл використовуються стандартні сферичні геометрії, на які накладалися текстурні карти високої роздільної здатності. Окрім базових дифузних текстур, застосовуються карти нормалей та карти висот. Це дає змогу ефективно

імітувати складний рельєф планет, кратери та гірські масиви, створюючи ілюзію деталізованої моделі при мінімальних витратах ресурсів системи. Особлива увага приділяється моделюванню джерел світла. Використання об’єкта PointLight, розташованого в умовному центрі системи, дозволяє відтворити природне падіння тіней та відблисків, що критично важливо для сприйняття глибини космічного простору.

Інтерактивність виступає головним фактором залучення користувача до освітнього процесу. Для реалізації вільного переміщення у просторі впроваджується модуль OrbitControls. Це забезпечує можливість обертання навколо об’єктів, динамічного масштабування та зміни кута огляду, що робить вивчення будови Сонячної системи наочним. Додатково було проаналізовано механізм Raycasting, який дозволяє ідентифікувати об’єкти, на які натискає користувач. Внаслідок цього стає можливим виведення контекстної інформації про кожну планету, перемикання між модулями новин та інтерактивними елементами інтерфейсу.

Оптимізація продуктивності виконується через керування циклом рендерингу за допомогою методу requestAnimationFrame. Це забезпечує стабільну частоту кадрів та плавність анімації руху небесних тіл по їхніх орбітах. Також було вирішено питання адаптивності: автоматичне оновлення матриці проекції камери при зміні розмірів вікна браузера гарантує коректне відображення моделей на будь-яких пристроях – від настільних ПК до смартфонів.

Для глибшого розуміння архітектури візуалізаційного модуля було систематизовано дані щодо використання компонентів бібліотеки Three.js, які подані в табл. 1.

У підсумку, застосування Three.js у розробці тематичних порталів дозволяє поєднувати інформативність з високим рівнем візуальної привабливості. Такий підхід перетворює статичне вивчення даних на динамічний досвід, де користувач стає активним дослідником віртуального всесвіту. Створення подібних рішень

підтверджує, що сучасні веб-інструменти є достатньо потужними для реалізації складних наукових та освітніх візуалізацій.

Таблиця 1 – Функціональні компоненти бібліотеки Three.js у структурі веб-порталу

№	Елемент архітектури	Технічна роль у Three.js	Вплив на візуальний результат
1	Scene	Контейнер для всіх 3D-об’єктів	Створення єдиного координатного простору для моделей
2	PerspectiveCamera	Моделювання точки огляду	Забезпечення ефекту глибини та реалістичної перспективи
3	WebGLRenderer	Візуалізація сцени у браузері	Виведення графіки на Canvas-елемент сторінки
4	SphereGeometry	Генерація геометрії планети	Формування базової кулястої форми небесного тіла
5	MeshStandardMaterial	Опис фізичних властивостей поверхні	Реалістичне поглинання та відбиття світлових променів
6	TextureLoader	Завантаження растрових карт	Надання об’єкту індивідуального вигляду конкретної планети
7	PointLight	Створення джерела світла (Сонце)	Формування динамічних тіней та яскравих відблисків
8	Clock / Delta Time	Керування часовими проміжками	Забезпечення рівномірної швидкості обертання об'єктів

Список використаних джерел:

1. Discover Three.js – The Missing Manual for Three.js. Повний посібник із засад роботи з тривимірним простором у браузері. URL: https://discoverthreejs.com/ (дата звернення: 15.03.2026).
2. MDN Web Docs. WebGL API: 2D and 3D graphics for the web. Технічний опис стандартів та низькорівневих можливостей обробки графіки. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API (дата звернення: 15.03.2026).
3. Three.js Documentation. Official website. Офіційний довідник із синтаксису, класів та методів бібліотеки. URL: https://threejs.org/docs/ (дата звернення: 16.03.2026).

4. Three.js Fundamentals. База знань із вирішення практичних завдань освітлення, камер та рендерингу. URL: https://threejsfundamentals.org/ (дата звернення: 16.03.2026).
5. WebGL Tutorial for Beginners – How to Create a 3D Scene. Навчальний матеріал щодо поетапної побудови віртуальних сцен. URL: https://www.freecodecamp.org/news/webgl-tutorial-for-beginners/ (дата звернення: 16.03.2026).

УДК 004.8:37

ВИКОРИСТАННЯ МЕРЕЖІ 5G НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОСВІТНИХ VR/AR-ДОДАТКІВ

Кондратенко Є.В.
Черкаський державний фаховий бізнес-коледж
Ратайчук П.Є
м. Черкаси, Україна

Сьогодні управління мережами 5G для мобільної VR/AR-освіти залежить від статичного розподілу ресурсів, якому бракує гнучкості для адаптації до змінних мережевих умов та різноманітних потреб користувачів. Нестабільність, піки затримки та жахливі візуальні ефекти ускладнюють навчання та підтримку зацікавленості. Для досягнення масштабного, безперервного та високоякісного імерсивного навчального досвіду нам потрібна інноваційна та адаптивна структура, яка може покращити продуктивність мережі в режимі реального часу на основі вхідних даних про якість досвіду.

Функції URLLC (Ultra-Reliable Low-Latency Communication) мереж 5G можуть допомогти задовольнити ці потреби . Завдяки високій пропускній здатності з’єднань 5G користувачі зможуть взаємодіяти з меншими затримками під час потокового передавання високоякісного контенту віртуальної та доповненої реальності. [1] Проте перед тим як велика кількість людей зможе повноцінно використовувати застосунки AR і VR, мережам 5G ще потрібно значно розвинутися, особливо для мобільних користувачів або тих, хто живе в місцях із нестабільним мережевим покриттям . На жаль, багато компаній досі

використовують застарілі методи розподілу мережевих ресурсів, які не встигають за швидкими змінами в мережах 5G та зростанням мережевого трафіку і нових вимог. Через цю негнучкість у VR- та AR-додатках для освіти можуть виникати проблеми з якістю користувацького досвіду (QoE), зокрема затримки, джитер (нерівномірність затримки) та зниження частоти кадрів. [3] Такі проблеми можуть ускладнювати процес навчання для студентів і зменшувати їхню зацікавленість.

Проблема стає значно серйознішою, коли системою користується велика кількість людей одночасно, наприклад у мобільних аудиторіях або під час імерсивних навчальних подій на рівні всього кампусу. Це пов’язано з тим, що мережевих ресурсів може не вистачати для всіх користувачів.

Мобільна система навчання VR/AR, що працює на базі мережі 5G та використовує мережеве сегментування (network slicing) з керуванням штучним інтелектом для забезпечення імерсивного та безперервного освітнього досвіду. [4] У системі використовується навчання з підкріпленням у реальному часі, яке постійно оптимізує мережеві ресурси, такі як затримка та пропускна здатність, щоб забезпечити максимально якісний віртуальний досвід. Викладачі можуть застосовувати цю технологію для створення гнучких та інтерактивних навчальних середовищ, а також використовувати аналітичні інструменти для відстеження успішності студентів. Крім того, система спрощує використання VR та AR у навчанні, забезпечуючи студентам швидке та гнучке підключення до мережі 5G. [2] На відміну від традиційних підходів статичного розподілу ресурсів, які використовують фіксовані параметри та не можуть адаптуватися до змінних умов мережі, ARL-NS оптимізує мережеве сегментування на основі даних QoE у реальному часі, використовуючи методи навчання з підкріпленням. Це дозволяє системі швидко реагувати на перевантаження мережі, зміну попиту користувачів та зміни в середовищі, забезпечуючи оптимальну роботу ресурсомістких застосунків, таких як VR та AR [1].

Порівняно з методами глибокого навчання, які потребують великих наборів даних і регулярного перенавчання, система ARL-NS краще підходить

для сценаріїв, що вимагають швидкої адаптації мережі. Евристичні методи можуть бути ефективними у контрольованих умовах, але менш ефективні в динамічних середовищах. На відміну від них, ARL-NS навчається на основі стану мережевого середовища та адаптує розподіл ресурсів у реальному часі, що підвищує масштабованість та ефективність системи. Крім того, ARL-NS може ефективніше підтримувати ширший спектр ресурсоємних застосунків, ніж стандартні підходи до мережевого сегментування, оскільки динамічно змінює розмір мережевих сегментів і розподіл ресурсів.

Представлено архітектуру ARL-NS, яка покращує якість користувацького досвіду (QoE) для інтерактивних освітніх VR/AR-застосунків у мобільному навчанні, оптимально використовуючи ресурси мережі 5G у режимі реального часу.

На рис. 1 зображено навчальну аудиторію, що використовує технологію 5G, у якій застосовуються віртуальна та доповнена реальність для покращення навчального процесу. [1]

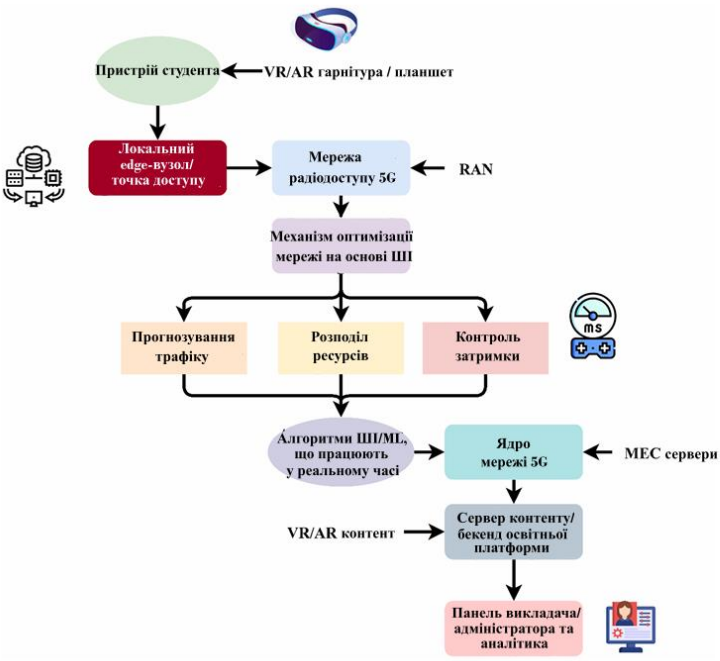


Рисунок 1 – Оптимізація мережі 5G на основі штучного інтелекту для захопливої освітньої віртуальної/доповненої реальності в мобільному навчанні

Пристрої студентів підключаються до мережі радіодоступу 5G (Radio Access Network, RAN) через найближчі edge-вузли або точки доступу. Спеціальний механізм оптимізації мережі, що використовує штучний інтелект і машинне навчання, керує затримкою, ефективно розподіляє ресурси та прогнозує мережевий трафік у реальному часі.

Після цього ядро мережі 5G та сервери Mobile Edge Computing (MEC) можуть швидше обмінюватися даними. Контент-сервер надає освітні VR та AR ресурси, що дозволяє студентам повністю зануритися в навчальний процес. [1] За допомогою аналітичних інструментів і інформаційних панелей адміністратори та викладачі можуть відстежувати рівень залученості та успішність студентів. Це дозволяє створювати гнучкі та сучасні навчальні середовища, які поєднують технології 5G та штучний інтелект.

Список використаних джерел:

1. Nyu Long, Weiheng Wang. AI-driven 5G network optimization for immersive educational VR/AR applications in mobile learning environments - Discover Internet of Things. *SpringerLink*. URL: https://link.springer.com/article/10.1007/s43926-025-00246-x (date of access: 08.03.2026).
2. Anastasiya Doroshenko, Kvitoslava Obelovska, Rostyslav Liskevych. Augmented and virtual reality for education: a case study in 5G network slicing⋆.
3. Leveraging 5G and AI Technologies to Enhance Real-Time. URL: https://onlinelibrary.wiley.com/doi/10.1002/itl2.70075 (date of access: 10.03.2026).
4. Kangdon Lee. The Future of Learning and Training in Augmented Reality. URL: https://scispace.com/pdf/the-future-of-learning-and-training-in-augmented-reality-3gdvtb10d1.pdf (date of access: 06.03.2026).

УДК 004.8

СИСТЕМА ІНТЕГРАЦІЇ ЧАТ-БОТІВ У СЕРВІСИ ДЛЯ АВТОМАТИЗАЦІЇ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ

Мельник А. О.
aartemko159@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В. Ю.
м. Черкаси, Україна

У сучасному цифровому середовищі чат-боти стають важливим інструментом автоматизації взаємодії між користувачами та інформаційними системами. Зростання обсягів онлайн-комунікацій і потреба у швидкому обслуговуванні клієнтів сприяють активному впровадженню чат-ботів у різні сфери, зокрема електронну комерцію, банківські сервіси, освіту та технічну підтримку.

Розвиток чат-ботів в Україні активно підтримується вітчизняними IT-компаніями, які створюють сучасні рішення на основі штучного інтелекту та обробки природної мови. Українські розробники успішно конкурують на світовому ринку та співпрацюють із міжнародними брендами.

Однією з провідних компаній є BotsCrew – українська компанія, заснована у Львові, що спеціалізується на створенні чат-ботів і голосових асистентів. Вона розробляє AI-рішення для бізнесу, автоматизує клієнтську підтримку та внутрішні процеси компаній. Компанія реалізувала понад 150 проєктів і співпрацює з відомими міжнародними брендами .

Ще одним прикладом є Master of Code Global – українська IT-компанія, яка входить до числа провідних світових розробників чат-ботів. Вона створює інтелектуальні системи для автоматизації обслуговування клієнтів і працює з великими міжнародними компаніями, такими як телекомунікаційні та фінансові організації .

Компанія AICoreBot спеціалізується на створенні корпоративних чат-ботів для навчання персоналу, автоматизації бізнес-процесів і внутрішніх комунікацій.

Її рішення дозволяють оптимізувати роботу компаній та підвищити ефективність управління знаннями .

Крім того, українські компанії активно впроваджують чат-боти у фінансовому секторі. Наприклад, ПриватБанк використовує ботів для обробки клієнтських запитів у різних каналах комунікації, що дозволяє автоматизувати до значної частини звернень користувачів .

Таким чином, українські компанії демонструють високий рівень розвитку технологій чат-ботів і штучного інтелекту. Вони не лише впроваджують інноваційні рішення на внутрішньому ринку, але й активно працюють на глобальному рівні, що підтверджує конкурентоспроможність української IT-галузі.

Чат-боти – це програмні системи, що використовують алгоритми обробки природної мови (NLP) та штучного інтелекту для імітації діалогу з користувачем. Вони можуть функціонувати як на основі заздалегідь визначених сценаріїв (rule-based), так і з використанням машинного навчання, що дозволяє їм адаптуватися до запитів користувачів і покращувати якість відповідей.

Основні компоненти чат-бота включають модуль обробки введення, механізм інтерпретації запиту, систему генерації відповіді та інтеграційний рівень для взаємодії з зовнішніми сервісами. Узагальнену модель роботи чат-бота можна представити у вигляді наступного співвідношення:

$$R = F(U, C) \quad (1), \text{ де}$$

R – відповідь чат-бота,

F – функція обробки запиту,

U – введення користувача,

C – контекст діалогу.

Для підвищення ефективності роботи застосовується збереження контексту розмови, що дозволяє формувати більш релевантні відповіді:

$$C_n = G(C_{n-1}, U_n) \quad (2), \text{ де}$$

C_n – оновлений контекст,

G – функція оновлення контексту,

U_n – новий запит користувача.

Інтеграція чат-ботів у сервіси передбачає їх взаємодію з базами даних, API та іншими інформаційними системами. Це дозволяє автоматизувати обробку запитів, надавати довідкову інформацію, виконувати транзакції та забезпечувати персоналізований досвід користувача. Зокрема, чат-боти можуть використовуватися для обробки замовлень, консультування клієнтів або підтримки користувачів у режимі реального часу.

Важливим аспектом є забезпечення безпеки даних при використанні чат-ботів. Для цього застосовуються сучасні протоколи передачі даних, механізми автентифікації користувачів та контроль доступу. Крім того, необхідно враховувати вимоги законодавства щодо захисту персональних даних, що регламентують обробку та зберігання інформації.

Таблиця 1 – Основні типи чат-ботів та їх характеристики

№	Тип чат-бота	Особливості	Сфера застосування
1	Rule-based	Працює за сценаріями	FAQ, підтримка
2	AI-бот	Використовує NLP	Консультації, асистенти
3	Гібридний	Комбінований підхід	Бізнес-сервіси
4	Голосовий	Обробка голосу	Смарт-пристрої

Процес інтеграції чат-бота у сервіси включає етапи аналізу потреб, розробки логіки діалогу, навчання моделі (для AI-ботів), тестування та впровадження. Після цього здійснюється постійний моніторинг та вдосконалення системи на основі зворотного зв’язку від користувачів.

Таким чином, чат-боти є ефективним інструментом автоматизації взаємодії з користувачами, що дозволяє підвищити швидкість обробки запитів, зменшити навантаження на персонал і покращити якість сервісу. Їх інтеграція у

сучасні інформаційні системи є важливим кроком у розвитку цифрових технологій та підвищенні конкурентоспроможності організацій.

Список використаних джерел:

1. Що таке чат-боти та як вони працюють - SendPulse Blog. URL: https://sendpulse.ua/blog/chatbots (дата звернення: 17.03.2026).
2. Natural Language Processing (NLP): основи та застосування - IBM. URL: https://www.ibm.com/topics/natural-language-processing (дата звернення: 17.03.2026).
3. What are chatbots? - Oracle. URL: https://www.oracle.com/chatbots/what-is-a-chatbot/ (дата звернення: 17.03.2026).
4. Artificial Intelligence in Customer Service - Salesforce. URL: https://www.salesforce.com/products/service-cloud/ai-customer-service/ (дата звернення: 17.03.2026).
5. Про захист персональних даних. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/go/2297-17 (дата звернення: 17.03.2026).
6. Chatbots in Business: Benefits and Use Cases - Tidio. URL: https://www.tidio.com/blog/chatbots/ (дата звернення: 17.03.2026).
7. Основи штучного інтелекту – Microsoft Learn. URL: https://learn.microsoft.com/uk-ua/training/paths/get-started-with-artificial-intelligence/ (дата звернення: 17.03.2026).
8. REST API: що це таке і як працює – RedHat. URL: https://www.redhat.com/uk/topics/api/what-is-a-rest-api (дата звернення: 17.03.2026).

УДК 004.77

МЕРЕЖЕВІ ТЕХНОЛОГІЇ В СИСТЕМІ «РОЗУМНОГО МІСТА»

Дем'яненко Д.В.
diamond26082006@gmail.com
Черкаський державний фаховий бізнес-коледж
Ратайчук П.Є.
м. Черкаси, Україна

Сучасний етап розвитку інформаційного суспільства характеризується активним впровадженням цифрових технологій у всі сфери життя людини. Одним із найбільш перспективних напрямів є концепція «розумного міста» (Smart City), яка передбачає інтеграцію інформаційно-комунікаційних технологій для підвищення ефективності міської інфраструктури, покращення якості життя населення та забезпечення сталого розвитку міських територій. Ключову роль у реалізації цієї концепції відіграють мережеві технології, що забезпечують передачу, обробку та зберігання великих обсягів даних між різними системами міської інфраструктури [1].

Мережеві технології у системі «розумного міста» базуються на використанні сучасних телекомунікаційних рішень, таких як Інтернет речей (IoT), бездротові мережі, хмарні обчислення та високошвидкісні канали передачі даних. Завдяки цим технологіям стає можливим об'єднання різноманітних міських сервісів у єдину інформаційну систему. До таких сервісів належать системи управління транспортом, енергопостачанням, освітленням, безпекою, екологічним моніторингом та комунальним господарством [2].

Одним із основних елементів мережевої інфраструктури «розумного міста» є технологія Інтернету речей. Вона передбачає використання великої кількості датчиків та пристроїв, які підключені до мережі та здатні автоматично передавати інформацію про стан навколишнього середовища або певних об'єктів. Наприклад, датчики можуть фіксувати рівень забруднення повітря, інтенсивність транспортного потоку, споживання електроенергії або заповненість паркувальних місць. Отримані дані передаються до

централізованих систем управління, де вони аналізуються та використовуються для прийняття управлінських рішень [2,3].

Важливу роль у забезпеченні ефективної роботи таких систем відіграють бездротові мережеві технології. До найбільш поширених належать Wi-Fi, LTE, 5G, а також спеціалізовані мережі для Інтернету речей, такі як LoRaWAN та NB-IoT. Використання цих технологій дозволяє забезпечити стабільне з'єднання між великою кількістю пристроїв та систем, що функціонують у межах міської інфраструктури. Зокрема, технологія 5G забезпечує високу швидкість передачі даних, низьку затримку сигналу та можливість підключення великої кількості пристроїв одночасно, що є надзвичайно важливим для функціонування систем «розумного міста» [3].

Ще одним важливим компонентом мережевих технологій у Smart City є використання хмарних платформ. Хмарні обчислення дозволяють ефективно зберігати та обробляти великі обсяги інформації, що надходить від різноманітних датчиків та інформаційних систем. Крім того, хмарні сервіси забезпечують масштабованість системи, що дає змогу поступово розширювати функціональні можливості міської інфраструктури без значних витрат на апаратне забезпечення [4].

Практичне застосування мережевих технологій у системі «розумного міста» можна спостерігати у багатьох країнах світу. Наприклад, у таких містах, як Сінгапур, Барселона та Амстердам, активно використовуються інтелектуальні транспортні системи, які дозволяють оптимізувати рух транспорту, зменшити затори та скоротити рівень забруднення повітря. Також впроваджуються системи «розумного» освітлення, що автоматично регулюють рівень освітленості в залежності від часу доби або інтенсивності руху на вулицях [5].

Незважаючи на значні переваги використання мережевих технологій у Smart City, існують і певні виклики. Серед основних проблем можна виділити питання кібербезпеки, захисту персональних даних, а також необхідність створення надійної та масштабованої мережевої інфраструктури. Оскільки системи «розумного міста» обробляють великі обсяги інформації, включаючи

дані про пересування людей, використання ресурсів та інші аспекти міського життя, важливо забезпечити високий рівень захисту цієї інформації від несанкціонованого доступу [4].

Таким чином, мережеві технології є ключовим елементом функціонування системи «розумного міста». Вони забезпечують взаємодію між різними компонентами міської інфраструктури, сприяють ефективному управлінню ресурсами та підвищенню якості життя населення. Подальший розвиток телекомунікаційних технологій та Інтернету речей відкриває широкі перспективи для розвитку концепції Smart City у майбутньому [1].

Список використаних джерел:

1. Bibri S. E., Krogstie J. Smart Cities: Foundations, Principles and Applications. Cham : Springer, 2022. 534 p.
2. A Review of Smart Cities Based on the Internet of Things Concept / S. Talari et al. Energies. 2020. Vol. 13, Iss. 2. P. 1–25
3. Internet of Things for Smart Cities: Technologies and Applications / A. Zanella et al. IEEE Internet of Things Magazine. 2021. Vol. 4, Iss. 1. P. 12–19.
4. Al-Turjman F., Lemayian J. Intelligence, Security, and IoT in Smart Cities. Future Generation Computer Systems. 2022. Vol. 129. P. 131–145.
5. World Smart Cities Outlook 2023: Digital Transformation of Cities / United Nations. New York : UN Publishing, 2023. 112 p.

УДК 004.738.5:378.147

ВИКОРИСТАННЯ ІНТЕРАКТИВНИХ ВЕБ-ТЕХНОЛОГІЙ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

Дорошенко В.М.
doroshenkovaleria2007@gmail.com
Черкаський державний фаховий бізнес-коледж
Немченко В.Ю.
м. Черкаси, Україна

У сучасних умовах цифровізації освіти інформаційні технології відіграють важливу роль у процесі навчання. Особливо актуальним є використання веб-технологій для організації інтерактивного навчання, оскільки вони дозволяють створювати зручні та доступні навчальні ресурси, що можуть використовуватися на різних пристроях.

Одним із перспективних напрямів є застосування інтерактивних веб-застосунків для вивчення іноземних мов. Такі застосунки дозволяють поєднати теоретичний матеріал із практичними завданнями, що сприяє кращому засвоєнню навчального матеріалу та підвищує мотивацію користувачів до навчання.

Для створення подібних навчальних ресурсів широко використовуються клієнтські веб-технології, зокрема HTML, CSS та JavaScript.

HTML використовується для формування структури веб-сторінок, CSS забезпечує оформлення інтерфейсу та адаптивність дизайну, а JavaScript дозволяє реалізувати інтерактивну логіку роботи застосунку. Використання JavaScript дає можливість створювати різні типи інтерактивних вправ, наприклад тести з вибором правильної відповіді, завдання на встановлення відповідності або вправи із заповненням пропусків.

Після виконання завдання користувач може отримувати миттєвий зворотний зв’язок щодо правильності відповіді, що значно підвищує ефективність навчання.

Інтерактивні веб-застосунки мають ряд переваг: доступність через браузер без необхідності встановлення додаткового програмного забезпечення, зручний

користувацький інтерфейс та можливість швидкого оновлення навчального контенту.

Таким чином, використання сучасних веб-технологій дозволяє створювати ефективні навчальні інструменти для вивчення англійської мови та інших дисциплін. Інтерактивні веб-застосунки можуть бути використані як додатковий засіб для самостійного навчання студентів та підтримки освітнього процесу.

Список використаних джерел:

1. Freeman A., Robson E. Head First HTML and CSS. O'Reilly Media, 2012.
2. Mozilla Developer Network. HTML, CSS and JavaScript Documentation. URL: https://developer.mozilla.org (дата звернення: 19.03.2026).
3. W3C. Web Standards and Technologies Overview. URL: https://www.w3.org/standards/ (дата звернення: 19.03.2026).

УДК 004.38

КВАНТОВІ КОМП'ЮТЕРИ: ПОТОЧНИЙ СТАН І ЗАСТОСУВАННЯ В ІТ ІНДУСТРІЇ

Драчук Д. Я.
dimondrachuk08@gmail.com
Черкаський державний фаховий бізнес-коледж
Люта М. В.
м. Черкаси, Україна

Квантовий комп'ютер – фізичний обчислювальний пристрій, функціонування якого ґрунтується на принципах квантової механіки, зокрема, принципі суперпозиції та явищі квантової сплутаності.

Квантовий комп'ютер відрізняється від звичайного комп'ютера зокрема тим, що класичний комп'ютер оперує даними, закодованими у двійкових розрядах (бітах), кожен з яких завжди перебуває в одному з двох станів (0 або 1), коли квантовий комп'ютер використовує квантові біти (кубіти), які можуть знаходитися у суперпозиції станів [1].

Стан квантових комп’ютерів на 2026 р. Станом на 2026 рік квантові комп'ютери активно переходять від стадії суто наукових експериментів до фази інженерної реалізації та інтеграції в реальні бізнес-процеси. Головний фокус індустрії змістився з простого нарощування кількості кубітів на забезпечення їхньої якості та виправлення помилок.

Поточний стан квантових технологій. Логічні кубіти замість фізичних: Найбільшим проривом останніх років стало створення «логічних» кубітів. Оскільки фізичні кубіти дуже нестабільні, компанії (Microsoft, Quantinuum, QuEra, IBM, Google) навчилися об'єднувати десятки фізичних кубітів в один стабільний логічний. Це відкриває шлях до надійних обчислень.

Гібридні обчислення: Квантові комп'ютери не замінять класичні. Вони працюють у тандемі (Quantum-Centric Supercomputing), де класичні системи виконують основну частину задач, а складні обчислення передаються квантовим процесорам.

Напрямки розвитку. Кібербезпека та криптографія: квантові алгоритми (зокрема алгоритм Шора) здатні зламати сучасні методи шифрування, що робить актуальним розвиток постквантової криптографії та впровадження нових стандартів безпеки. Квантовий розподіл ключів (QKD): використання законів квантової фізики дозволяє створювати захищені канали зв’язку, де будь-яке втручання може бути виявлене.

Машинне навчання та штучний інтелект: квантові технології відкривають нові можливості для оптимізації навчання моделей та аналізу великих даних.

Складні задачі оптимізації: квантові алгоритми застосовуються у логістиці, фінансах і виробництві для розв’язання задач із великою кількістю варіантів, що значно перевищують можливості класичних систем.

Розробка програмного забезпечення: формується новий напрям – квантова розробка (Quantum Software Engineering), що передбачає використання спеціалізованих інструментів, таких як Qiskit, Cirq та Q#.

Квантові комп'ютери більше не є далекою науковою фантастикою – це реальний інструмент, який уже трансформує IT-індустрію. Хоча вони не

замінять класичні комп'ютери у повсякденних завданнях, їхня здатність вирішувати надскладні задачі створює нові можливості розвитку технологій. Головним викликом залишається адаптація інфраструктури до «квантової епохи», особливо у сфері кібербезпеки [2].

Список використаних джерел:

1. Що таке квантові комп’ютери. Cityhost. URL: https://cityhost.ua/uk/blog/scho-take-kvantovi-komp-yuteri.html (дата звернення: 23.03.2026).
2. Застосування квантових комп’ютерів. Colobridge Blog. URL: https://blog.colobridge.net/uk/2025/05/applications-of-quantum-computing-ua/ (дата звернення: 23.03.2026).

УДК 004.312:378.147

МОДЕЛЮВАННЯ СИСТЕМ АВТОМАТИЗАЦІЇ РОЗУМНОГО БУДИНКУ НА ОСНОВІ IOT-ТЕХНОЛОГІЙ У СЕРЕДОВИЩІ CISCO PACKET TRACER

Ільєнко О.М.
sashailienko555@gmail.com
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м.Черкаси, Україна

Сучасний етап розвитку інформаційних технологій жваво сприяє інтеграції автоматизованих систем у повсякденне життя суспільства. Одним із ключових напрямків цього процесу є технологія Інтернету речей (Internet of Things, IoT), що передбачає створення інтегрованих мереж, які об’єднують різноманітні пристрої з метою збору, передачі та оброблення даних. Одним із найбільш поширених напрямів практичного застосування технологій IoT є концепція «розумного будинку». Такі системи забезпечують автоматизацію управління освітленням, безпековими механізмами, а також іншими елементами побутової інфраструктури. Інтеграція подібних рішень сприяє підвищенню рівня комфорту

і безпеки проживання, а також оптимізації енерговитрат, що спрямовано на підвищення загальної енергоефективності житлових приміщень.

У концепції розумного будинку всі пристрої об’єднані в єдину IoT-систему, в якій взаємодія реалізується за принципом «датчик – контролер – виконавчий пристрій». Центральним елементом є IoT-контролер, який забезпечує підключення до мережі та координацію роботи всіх компонентів системи. До контролера через бездротові протоколи підключаються датчики та виконавчі пристрої. Датчики руху виконують функцію збору вхідних даних і, у разі виявлення активності, передають відповідний сигнал на контролер. Після обробки отриманої інформації відповідно до заданих сценаріїв контролер формує керуючі команди для виконавчих пристроїв. Зокрема, розумні лампи отримують сигнал від контролера та автоматично вмикаються, вимикаються або змінюють параметри освітлення, тоді як електронні двері або розумні замки можуть як виконувати команди відкриття або блокування, так і передавати зворотний статус системі. Таким чином, IoT-контролер забезпечує узгоджену взаємодію між усіма елементами системи, реалізацію автоматизованих сценаріїв і підвищення рівня безпеки, енергоефективності та комфорту в розумному будинку.

Однією з ключових переваг систем розумного будинку є їхня здатність забезпечувати автоматизацію широкого спектра процесів, що дозволяє мінімізувати участь користувача в повсякденному керуванні пристроями та підвищити ефективність функціонування всієї системи.

Для моделювання та розробки систем розумного будинку й IoT застосовуються різні програмні середовища, кожне з яких має свої особливості та сферу використання. Одним із найбільш комплексних інструментів є Cisco Packet Tracer, який дозволяє не лише моделювати IoT-пристрої, а й проєктувати повноцінну мережеву інфраструктуру з можливістю програмування та візуалізації середовища. Для тестування мікроконтролерів, таких як ESP32 або Arduino, ефективним є Wokwi – онлайн-середовище, що підтримує WiFi-симуляцію та забезпечує високу сумісність коду з реальним обладнанням. У

свою чергу, Node-RED використовується для створення логіки взаємодії між пристроями за допомогою візуального програмування потоків даних, що значно спрощує інтеграцію з різними сервісами

Таблиця 1 – Порівння середовищ

№	Cisco Packet Tracer	Комплексне моделювання IoT та мереж розумного будинку з підтримкою програмування і топологій	Середній / Високий
1	Wokwi	Тестування коду мікроконтролерів (ESP32, Arduino) з можливістю WiFi-симуляції	Низький / Середній
2	Node-RED	Побудова логіки роботи розумного будинку та інтеграція сервісів через візуальні потоки	Середній
3	Tinkercad Circuits	Базове навчання роботі з електронікою та датчиками у простому середовищі	Дуже низький
4	CupCarbon	Моделювання бездротових сенсорних мереж та Smart City з аналізом сигналів і геолокацією	Високий

Для початкового ознайомлення з електронікою доцільно застосовувати Tinkercad Circuits, який вирізняється простотою та наочністю, але має обмежені можливості для складних IoT-проєктів. Окремо варто відзначити CupCarbon – спеціалізований інструмент для моделювання бездротових сенсорних мереж і систем типу Smart City, що дозволяє аналізувати радіопокриття та енергоспоживання, проте потребує більш глибоких технічних знань. Таким чином, вибір середовища залежить від поставлених завдань, рівня підготовки користувача та необхідного рівня деталізації моделювання.

Подальша практична реалізація описаних підходів може бути ефективно змодельована в середовищі Cisco Packet Tracer, яке забезпечує підтримку IoT-пристроїв і дозволяє створювати цілісні моделі розумного будинку. У межах даної платформи реалізується налаштування логіки взаємодії між компонентами системи, розробка сценаріїв автоматизації та імітація обміну даними з використанням різних мережевих протоколів. Це, у свою чергу, забезпечує можливість відтворення умов, наближених до реальних, проведення тестування

працездатності системи та оцінки ефективності прийнятих рішень без залучення фізичного обладнання, що є важливим етапом у процесі проєктування IoT-систем.

Список використаних джерел:

1. Cisco Systems. Cisco Packet Tracer – Networking Simulation Tool**.** – Режим доступу: https://www.netacad.com
2. Cisco Networking Academy. Introduction to IoT. – Cisco Networking Academy, URL:https://www.netacad.com/catalogs/learn
3. APNIC – What is the Internet of Things URL:https://blog.apnic.net/tag/internet-of-things

УДК 004.8:004.912

ЧАТ-БОТИ ТА ВІРТУАЛЬНІ АСИСТЕНТИ В ОБСЛУГОВУВАННІ КЛІЄНТІВ: ПРИНЦИПИ ОБРОБКИ ТЕКСТОВОЇ ІНФОРМАЦІЇ

Кисла Л. С.
kislaliliya0@gmail.com
Черкаський державний фаховий бізнес-коледж
Злочевська-Краснощок Д. С.
м.Черкаси, Україна

У сучасному світі технологій штучний інтелект активно застосовують у різних сферах діяльності. Однією з найважливіших сфер використання таких технологій є обслуговування клієнтів. Компанії прагнуть покращувати свою роботу, швидко відповідати на запити користувачів, надавати інформацію про товари та послуги і забезпечувати якісну підтримку. Саме тому дедалі частіше використовуються чат-боти та віртуальні асистенти.

Чат-бот – це програмна, яка здатна імітувати спілкування з користувачем за допомогою текстових або голосових повідомлень. Чат-боти можуть бути створені для різних цілей, таких як обслуговування клієнтів, пошук інформації, персональна допомога та для розважальних цілей. Вони можуть бути інтегровані в месенджери, веб-сайти або мобільні додатки, дозволяючи користувачам

взаємодіяти з ними за допомогою текстових повідомлень або голосових команд [1].

Чат-боти побудовані на основі обробки природної мови (Natural Language Processing, NLP), яка дозволяє машинам розуміти людську мову, брати участь у розмовах і виконувати прості автоматизовані завдання [2]. Також вони використовують аналітику й аналіз, щоб передбачати потреби користувача та надавати актуальні рекомендації.

Обробка природної мови – це одна з можливостей ШІ, яку використовують чат-боти, щоб аналізувати значні обсяги даних. Процес влаштований так, людина вводить текст, після чого ШІ розшифровує його значення та генерує й надає відповідну відповідь [3].

Процес обробки текстової інформації у чат-ботах складається з кількох етапів. Спочатку система отримує текстове повідомлення користувача. Потім виконується попередня обробка тексту, ШІ аналізує введений фрагмент, щоб визначити ціль чи намір користувача. Після цього відбувається генерація відповіді та завершальному етапі відповідь надсилається користувачу.

В компаніях використовуються два типи чат-ботів на основі штучного інтелекту: транзакційні та розмовні. Обидва типи виконують завдання, але відрізняються складністю та способом використання.

Таблиця 1 – Етапи обробки інформації чат-ботом

№	Етап	Опис процесу
1	Отримання повідомлення	Система приймає повідомлення від користувача
2	Обробка тексту	Аналіз введеного тексту, визначення що саме хоче користувач
3	Пошук відповіді	Пошук інформації для формування відповіді
4	Надсилання відповіді	Передача сформованого повідомлення користувачу

Транзакційні чат-боти генерують автоматичні відповіді у форматі розмови. Взаємодія з цими чат-ботами є високоструктурованою та конкретною. Тому такі чат-боти є дуже корисними для компаній, які заздалегідь знають, з якими

проблемами можуть зіштовхнутися клієнти. Наприклад, ресторани, служби доставки та банки використовують транзакційні чат-боти, щоб обробляти поширені запитання клієнтів [3].

Розмовний чат-бот – це складніший варіант, який забезпечує персоналізовану взаємодію. Такий тип чат-ботів використовує розмовний ШІ, щоб відповідати на повідомлення користувача як людина.

Розмовні чат-боти використовують ШІ, обробку природної мови, бази знань і відповідну контекстну інформацію, щоб виявляти нюанси в запитаннях та відповідях користувача й надавати актуальні відповіді як людина. Ці чат-боти на основі ШІ враховують контекст і постійно використовують розуміння та обробку природної мови, а також машинне навчання, щоб увесь час удосконалюватися [3].

Компанії застосовують чат-боти як віртуальних помічників для обробки запитів у службі підтримки клієнтів або для допомоги працівникам. Завдяки використанню чат-ботів компанії покращують обслуговування та пришвидшують продажі.

Чат-боти на основі штучного інтелекту надають клієнтам допомогу та підтримку за запитом і без обмежень. Під час взаємодії з чат-ботами клієнти отримують відповіді на запитання в будь-який час. Також чат-боти можуть спростити взаємодію під час продажів і допомогти клієнтам сформувати зв’язок із брендами [3].

Сучасні чат-боти та віртуальні асистенти використовуються у різних сферах діяльності для підвищення ефективності та якості обслуговування. Вони допомагають автоматизувати процеси, скоротити час реакції на запити користувачів та забезпечують безперервну підтримку.

Таблиця 2 – Сфери застосування чат-ботів

№	Сфера використання	Приклади застосування чат-ботів на основі ШІ
1	Служба підтримки клієнтів	Чат-боти виконують роль віртуальних агентів, доступних цілодобово, які приймають та обробляють запити та виконують стандартні завдання.
2	Відділ кадрів	Допомагають керувати пільгами працівників, повідомляють про зміни в політиці, спрощують подання заяв на лікарняний або відпустку.
3	Фінанси та бухгалтерський облік	Оновлюють дані про постачальників, формують фінансові звіти, відкривають запити на закупівлі.
4	Маркетинг	Надсилають персоналізовані пропозиції клієнтам та сприяють підвищенню їх залучення.
5	Продажі	Проводять відбір клієнтів, надають пропозиції та звільняють час продавців.

Використання чат-ботів має багато переваг. Вони можуть працювати цілодобово без перерв, здатні одночасно обслуговувати велику кількість користувачів та дозволяють зменшити витрати компаній на підтримку клієнтів.

Чат-боти сприяють підвищенню роботи співробітників, автоматизуючи рутинні завдання та забезпечуючи швидкий доступ до необхідної інформації. Це дозволяє працівникам зосередитися на більш складних та стратегічно важливих завданнях. Вони також допомагають компаніям збирати та аналізувати дані про запити клієнтів, що дозволяє оптимізувати процеси.

В сучасному світі впровадження чат-ботів і віртуальних асистентів є важливим кроком, оскільки поєднує автоматизацію, підвищення продуктивності та покращення обслуговування клієнтів, роблячи бізнес більш адаптованим до потреб цифрового середовища та клієнтів.

Список використаних джерел:

1. Білогрудов Д. В. РОЗРОБКА ЧАТ-БОТУ В TELEGRAM З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ: автореф. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ на курсову роботу. Київ, 2023. 38 с. URL: https://ekmair.ukma.edu.ua/server/api/core/bitstreams/d57efa11-8cd8-411a-a915-50d3c29e0c81/content (дата звернення: 13.03.2026).

2. Азиранкулов Є. А. Спосіб ідентифікації використання чат-ботів зі штучним інтелектом при написанні студентами програм мовою С : автореф. Магістирська десертація. Київ, 2024. 99 с. URL: https://ela.kpi.ua/server/api/core/bitstreams/29c0e561-ab14-455f-9504-413617e3448b/content (дата звернення: 13.03.2026).
3. Чат-боти на основі ШІ Microsoft Copilot. Microsoft – AI, Cloud, Produktivität, Computing, Gaming und Apps. URL: https://www.microsoft.com/uk-ua/microsoft-copilot/copilot-101/ai-chatbot?utm_source (дата звернення: 13.03.2026).

СЕКЦІЯ 4

РОБОТОТЕХНІКА ТА АДМІНІСТРУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ

УДК 004.7:621.311.1

GREEN IT У МЕРЕЖЕВІЙ ІНФРАСТРУКТУРІ ПІДПРИЄМСТВА

Волошко М.В.
voloshko.maks@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р.В.
м. Черкаси, Україна

Сучасні офісні середовища характеризуються високим рівнем залежності від інформаційно-комунікаційних технологій, що зумовлює значне споживання електричної енергії мережевою інфраструктурою. До її складу входять маршрутизатори, комутатори, сервери, точки бездротового доступу, системи безперебійного живлення та інше допоміжне обладнання. Зростання кількості цифрових сервісів і постійне функціонування мережевих пристроїв призводять до збільшення енергетичних витрат, тому питання оптимізації енергоспоживання набуває важливого практичного значення для сучасних організацій.

На відміну від звичайних електронних пристроїв, мережеве обладнання часто працює цілодобово незалежно від рівня активності користувачів. У більшості офісів навантаження на мережу суттєво змінюється протягом доби: у робочі години спостерігається максимальна активність, тоді як у нічний час або у вихідні дні значна частина ресурсів використовується лише частково. Проте обладнання продовжує функціонувати у стандартному режимі, що призводить до надлишкового споживання електроенергії [1].

Одним із ключових напрямів оптимізації є використання енергоефективного мережевого обладнання. Сучасні комутатори та маршрутизатори підтримують технології енергозбереження, зокрема Energy Efficient Ethernet (EEE), які дозволяють автоматично знижувати енергоспоживання при зменшенні навантаження на канали зв’язку. Використання таких пристроїв дає можливість скоротити витрати електроенергії без негативного впливу на продуктивність мережі [2].

Важливим аспектом є також оптимізація серверної інфраструктури. У багатьох організаціях окремі сервери використовуються неефективно, працюючи із низьким рівнем завантаження процесора та пам’яті. У таких умовах доцільним є впровадження технологій віртуалізації, які дозволяють об’єднати кілька сервісів на одному фізичному сервері. Це забезпечує зменшення кількості обладнання, скорочення енергоспоживання та спрощення адміністрування системи.

Для ефективного управління енергоспоживанням необхідним є постійний моніторинг роботи мережевої інфраструктури. Використання спеціалізованих систем моніторингу дозволяє аналізувати рівень навантаження, температуру обладнання, споживання електроенергії та виявляти компоненти, що працюють неефективно. Такі системи дають змогу адміністратору оперативно реагувати на перевантаження або надмірні витрати ресурсів[3].

Особливу роль відіграє автоматизація керування живленням обладнання. У сучасних офісах доцільно впроваджувати політики автоматичного переведення окремих пристроїв у режим енергозбереження або їхнього вимкнення у неробочий час [4]. Наприклад, додаткові точки доступу чи резервні комутатори можуть автоматично деактивуватись у періоди мінімальної активності користувачів. Такий підхід дозволяє зменшити непродуктивне використання електроенергії [5].

Значний вплив на енергоефективність має правильна побудова мережевої топології. Надлишкові мережеві вузли, дублювання обладнання та нераціональне розміщення пристроїв збільшують енергетичні витрати та ускладнюють адміністрування. Оптимізація структури мережі, централізація управління та скорочення кількості проміжних пристроїв сприяють підвищенню ефективності функціонування всієї інфраструктури. Топологію енергоефективної офісної мережі показано на рис.1.

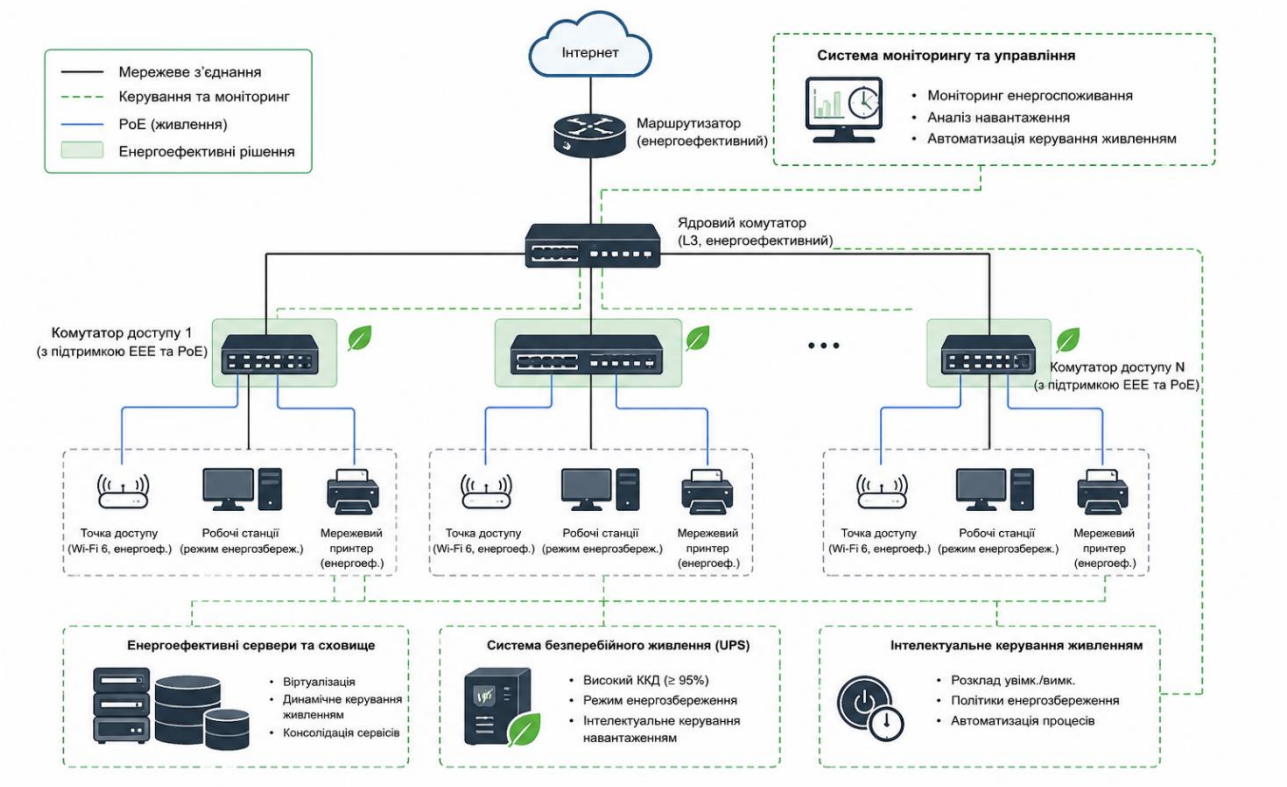


Рисунок 1 – Топологія енергоефективної офісної мережі

Не менш важливим є використання сучасних систем безперебійного живлення (UPS). Застарілі UPS часто мають низький коефіцієнт корисної дії та створюють додаткові втрати електроенергії. Використання сучасних моделей із високим ККД та функціями енергозбереження дозволяє мінімізувати втрати при перетворенні електроенергії та підвищити загальну надійність мережевої інфраструктури.

У практичному аспекті оптимізація енергоспоживання повинна здійснюватися комплексно та включати технічні, програмні й організаційні заходи. Поєднання модернізації обладнання, впровадження автоматизованого моніторингу та оптимізації режимів роботи мережі дозволяє суттєво знизити витрати електроенергії, підвищити стабільність роботи інфраструктури та зменшити тепловиділення обладнання.

Таким чином, оптимізація енергоспоживання мережевої інфраструктури офісу є важливим напрямом розвитку сучасних інформаційних систем. Раціональне використання енергетичних ресурсів дозволяє не лише скоротити

фінансові витрати організації, а й підвищити ефективність роботи обладнання, покращити надійність мережі та забезпечити більш екологічний підхід до експлуатації ІТ-інфраструктури.

Список використаних джерел:

1. Бондаренко В. О. Енергоефективність мережевої інфраструктури підприємств в умовах цифровізації. Комп'ютерні системи та мережі. 2024. № 3. С. 41–49.
2. Коваленко І. М. Сучасні підходи до оптимізації енергоспоживання серверного обладнання. Вісник комп'ютерної інженерії. 2023. Вип. 12. С. 88–95.
3. Мельник О. М. Адміністрування та енергоефективність інформаційних систем : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2024. 296 с.
4. Петренко В. І. Моніторинг та управління енергоспоживанням мережевої інфраструктури. Інформаційні технології та системи. 2025. № 1. С. 73–81.
5. Brown T., Wilson J. Energy-Efficient Network Infrastructure Management. 2nd ed. New York: Springer, 2025. 385 p.

УДК 004.8:004.056

ZERO TRUST ARCHITECTURE ЯК СУЧАСНА КОНЦЕПЦІЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ КОРПОРАТИВНИХ МЕРЕЖ

Ковальчук В.М.
vladislav72k@gmail.com
Черкаський державний фаховий бізнес-коледж
Захарова М.В.
м. Черкаси, Україна

Сучасні корпоративні мережі стикаються з безпрецедентним зростанням кіберзагроз, що ставлять під сумнів ефективність традиційних методів захисту. Стандартні моделі безпеки, засновані на довірі до внутрішньої мережі та зонах контролю, поступово втрачають актуальність через поширення хмарних сервісів,

віддаленої роботи та особистих пристроїв працівників, що використовуються для доступу до корпоративних ресурсів. Альтернативою класичному підходу є концепція Zero Trust Architecture (ZTA), яка базується на припущенні, що мережа вже потенційно скомпрометована.

Метою цієї роботи є дослідження архітектури Zero Trust як сучасної концепції забезпечення безпеки корпоративних мереж, аналіз її основних принципів, компонентів і механізмів контролю доступу.

Традиційні підходи до мережевої безпеки базуються на концепції «периметра», де захист будується навколо корпоративної мережі, формуючи умовну межу між довіреним внутрішнім середовищем і потенційно небезпечним зовнішнім. Така модель передбачає використання механізмів контролю доступу для користувачів, застосунків та інших компонентів, що взаємодіють із ресурсами. Її ключове припущення полягає в тому, що загрози походять переважно ззовні, тоді як внутрішнім суб'єктам можна довіряти. Однак із розвитком цифрових технологій і поширенням дистанційної роботи ці межі стають розмитими, що знижує ефективність периметрового підходу.

На відміну від цього, модель Zero Trust зосереджується безпосередньо на захисті ресурсів. У ній мережеве розташування не визначає рівень довіри: жоден користувач, пристрій чи сервіс не вважається надійним за замовчуванням. Доступ надається динамічно – лише після автентифікації, авторизації та оцінки контексту кожного запиту.

ZTA використовує принципи нульової довіри для планування та захисту корпоративної інфраструктури та робочих процесів. За дизайном середовище ZTA охоплює поняття відсутності неявної довіри до активів і суб'єктів, незалежно від їхнього фізичного чи мережевого розташування. Таким чином, ZTA ніколи не надає доступ до ресурсів, доки предмет, актив або робоче навантаження не будуть перевірені. Є багато логічних компонентів, які складають розгортання ZTA на підприємстві. Ці компоненти можуть працювати як локальна служба або через хмарну службу.

Таблиця 1 – Порівняння моделей безпеки

№	Захист «Периметр»	Zero Trust
1	Довіра за замовчуванням	Верифікація є обов’язковою
2	Захист фокусується на периметрах контрольованої зони	Захист фокусується на ресурсах
3	Доступ надається на основі статичних рішень	Доступ надається на основі динамічної політики в доступ є бінарним
4	Не передбачається обов'язкове використання	Передбачається обов'язкове використання політик політик розмежування доступу
5	Ручне управління ризиками	Автоматизоване управління ризиками

Модель концептуальної основи на рис.1 показує базовий зв’язок між компонентами та їх взаємодією. Точка прийняття рішень (PDP) розбивається на два логічні компоненти: механізм політики та адміністратор політики. Логічні компоненти ZTA використовують окрему площину керування для зв’язку, тоді як дані програми передаються на площині даних.

У моделі Zero Trust Architecture точка прийняття рішень (Policy Decision Point, PDP) є центральним логічним компонентом, що визначає можливість доступу користувача або пристрою до ресурсу. PDP складається з двох частин: двигуна політики (Policy Engine, PE) та адміністратора політики (Policy Administrator, PA).

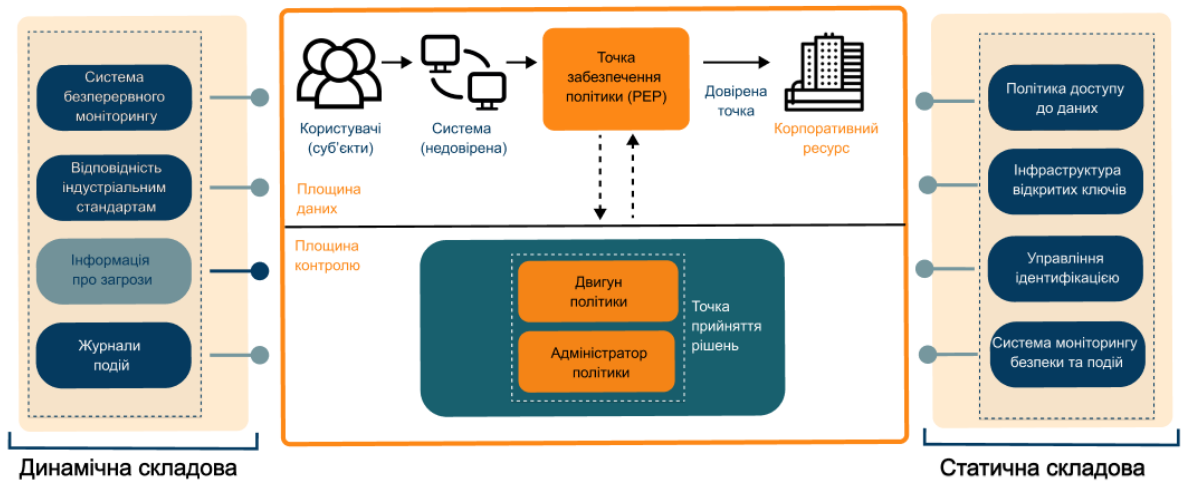


Рисунок 1 – Логічні компоненти Zero Trust

PE виконує аналітичну функцію, оцінюючи запит на основі політик безпеки, ідентичності користувача, стану пристрою та контексту підключення, після чого формує рішення про дозвіл або відмову. PA забезпечує виконання цього рішення, встановлюючи або розриваючи з'єднання, а також керуючи видачею тимчасових облікових даних чи токенів доступу.

Ядро Zero Trust Architecture використовує дані з різних джерел для контролю мережевих з'єднань і складається зі статичної та динамічної складових. До статичної належать політики доступу, інфраструктура відкритих ключів (PKI), системи керування ідентифікацією та механізми відповідності вимогам. Динамічна складова включає системи безперервної діагностики (CDM), розвідки загроз, журнали активності та системи SIEM, які забезпечують моніторинг і оцінювання рівня безпеки.

Обробка запитів у PDP здійснюється за допомогою інтелектуального механізму політик (IPE), що поєднує статичні й динамічні правила. У межах Zero Trust мережа поділяється на окремі площини, зокрема площину даних, яка відповідає за передачу трафіку між користувачем і ресурсами.

Процес автентифікації в Zero Trust виходить за межі перевірки ідентичності та враховує стан пристрою, мережевий контекст і поведінку користувача. Вона охоплює як автентифікацію користувача, так і пристрою, використовуючи, зокрема, біометричні методи та автентифікацію на фізичному рівні (PLA), що підвищує достовірність доступу.

У результаті проведеного аналізу встановлено, що Zero Trust Architecture формує новий підхід до захисту корпоративних мереж, орієнтований на контроль доступу з урахуванням контексту, стану пристроїв і поведінкових характеристик користувачів. Використання динамічних політик, безперервної автентифікації та моніторингу дозволяє адаптувати рішення відповідно до поточного рівня ризику. Поєднання статичних правил із динамічними даними забезпечує більш гнучке та ефективне управління доступом порівняно з традиційними моделями. Впровадження ZTA підвищує стійкість інформаційних систем до сучасних

кіберзагроз і забезпечує контрольовану взаємодію між користувачами, пристроями та ресурсами.

Список використаної літератури:

1. Rose S., Borchert O., Mitchell S., Connelly S. Zero Trust Architecture. Gaithersburg, MD: National Institute of Standards and Technology (NIST), 2020. 59 p. (NIST Special Publication 800-207). Режим доступу: Zero Trust Architecture (NIST SP 800-207) (дата звернення: 07.04.2026).
2. Zero Trust Architecture / C. Green-Ortiz et al. Cisco Press, 2022.
3. Introductory Guide to Zero Trust Architecture. Zero Trust Education, 2024.
4. Zero Trust Security Paradigm: A Comprehensive Survey and Research Analysis / Journal of Electrical Systems. 2023. No19(2). С. 28–37.
5. Verify and trust: A multidimensional survey of zero-trust security in the age of IoT / [M. A. Azad, S. Abdullah, J. Arshad та ін.] // Internet of Things. 2024. No27. С. 101227.
6. Exploring Effective Zero Trust Architecture for Defense Cybersecurity: A Study / [Y. Kim, S. Sohn, K. T. Kim та ін.] // KSII Transactions on Internet and Information Systems. 2024. No18(9). С. 2665–2691.

УДК 004.4

DEVOPS - ПІДХОДИ В УПРАВЛІННІ РОБОТОТЕХНІЧНИМИ СИСТЕМАМИ

Мусієнко О.О.
musienkos1975@gmail.com
Черкаський державний фаховий бізнес-коледж
Литовченко В.О.
м. Черкаси, Україна

Сьогодні робототехнічні системи стають все складнішими. Вони поєднують у собі апаратну та програмну частини і часто працюють у режимі реального часу. Через це зростають вимоги до їх розробки, тестування та підтримки. Звичайні підходи не завжди добре справляються з такими задачами, тому все частіше використовують DevOps-підходи [2].

DevOps – це підхід, який поєднує процеси розробки, тестування, розгортання та підтримки програмного забезпечення. Його головна ідея – автоматизувати ці процеси та зробити їх швидшими. Важливу роль відіграють безперервна інтеграція (Continuous Integration, CI) та безперервна доставка (Continuous Delivery, CD), які дозволяють швидко перевіряти зміни в програмі [1].

До основних складових DevOps, які можна застосовувати в робототехніці, належать: безперервна інтеграція, безперервне розгортання, контейнеризація, моніторинг і логування, а також DevSecOps, який враховує питання безпеки [4].

На рисунку 1 показано, як DevOps може використовуватися в робототехнічній системі на прикладі FPV-дрона. У цій моделі всі процеси пов’язані між собою і працюють разом.

Рисунок 1 – Схема застосування DevOps у робототехнічній системі FPV-дрона

У лівій частині моделі знаходяться процеси розробки та тестування. Тут використовується CI/CD-конвеєр, який дозволяє автоматично перевіряти зміни у програмному коді. Контейнеризація допомагає зробити однакові умови для роботи програм [3].

У центрі моделі знаходиться сам FPV-дрон, який виконує основні функції. Він отримує команди, обробляє дані та взаємодіє з іншими частинами системи.

Права частина моделі відповідає за моніторинг і аналіз даних. Система отримує телеметрію в реальному часі та дозволяє швидко реагувати на проблеми.

Також у моделі використовується хмарна інфраструктура, яка забезпечує зберігання даних і керування системою. Частина обробки виконується безпосередньо на пристрої завдяки edge-обчисленням, тобто коли дані обробляються не в хмарі, а прямо на самому пристрої. Це дозволяє швидше отримувати результат і зменшує затримки [5].

Щоб показати ефективність такого підходу, було порівняно роботу системи до і після використання DevOps (рис. 2).

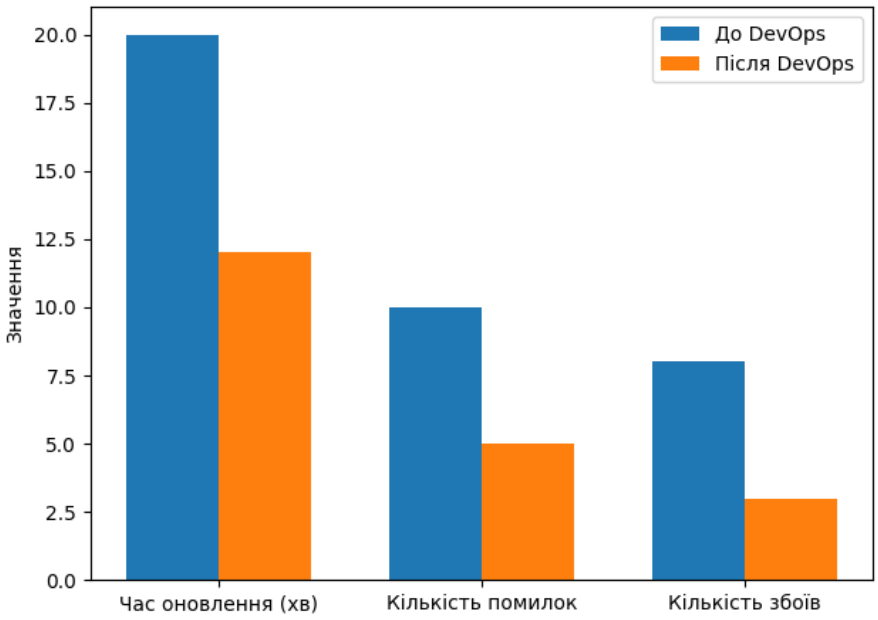


Рисунок 2 – Порівняння ефективності роботи системи до та після впровадження DevOps-підходів

З діаграми видно, що після використання DevOps зменшується час оновлення програмного забезпечення, а також знижується кількість помилок. Це свідчить про підвищення стабільності роботи системи.

Отже, DevOps-підходи допомагають швидше оновлювати програмне забезпечення, зменшують кількість помилок і роблять систему більш надійною,

а також зручні при збільшенні кількості пристроїв. Таким чином, їх доцільно використовувати у робототехнічних системах, оскільки запропонована модель може бути основою для створення гнучких і надійних систем.

Список використаних джерел

1. Humble J., Farley D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. – Boston: Addison-Wesley, 2021. – 512 p.
2. AWS. What is DevOps? 2024. URL: https://aws.amazon.com/devops/what-is-devops/ (дата звернення: 16.03.2026).
3. Docker Inc. What is a Container? 2024. URL: https://www.docker.com/resources/what-container/ (дата звернення: 17.03.2026).
4. Red Hat. What is DevOps? 2023. URL: https://www.redhat.com/en/topics/devops (дата звернення: 18.03.2026).
5. Microsoft. DevOps overview. 2024. URL: https://learn.microsoft.com/en-us/devops/what-is-devops (дата звернення: 18.03.2026).

УДК 004.7

ІНФОКОМУНІКАЦІЙНА МЕРЕЖА ЯК ОСНОВА АДМІНІСТРУВАННЯ СУЧАСНИХ РОБОТОТЕХНІЧНИХ КОМПЛЕКСІВ

Федоренко Д. В.
fedorenkobmx@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р.В.
м. Черкаси, Україна

Інфокомунікації є порівняно новим терміном, що відображає фундаментальну парадигму сучасного інформаційного суспільства – нерозривний зв’язок інформаційних і телекомунікаційних елементів інформаційного обміну. Дана парадигма розвивається в умовах глибокої

конвергенції, тобто взаємного проникнення та інтеграції різноманітних технологій, що призводить до формування якісно нових властивостей систем передачі та обробки даних.

У рамках цієї парадигми інфокомунікації представляють собою синтез телекомунікацій з інформаційними, комп'ютерними технологіями та радіотехнологіями [1]. Вони забезпечують не лише транспортування сигналів на відстань, але й ідентифікацію інформаційного змісту, його оптимальну обробку, маршрутизацію, перетворення та програмне управління.

Під інфокомунікаціями науково розуміють сукупність засобів обробки, накопичення, зберігання інформації та її перенесення в просторі, імплементованих у єдину мережну структуру. Завдяки такій інтегрованій архітектурі досягається висока доступність інформаційних ресурсів і ефективний, надійний інформаційний обмін між усіма учасниками системи незалежно від їх територіального розташування.

Інфокомунікаційна мережа визначається як складна, територіально розосереджена сукупність інформаційних і обчислювальних ресурсів, програмних комплексів управління, які розміщуються як у кінцевих системах мережі, так і в термінальних системах користувачів [1]. Взаємодія між цими компонентами забезпечується засобами телекомунікацій, унаслідок чого утворюється єдина мультисервісна платформа, здатна надавати різноманітні послуги в реальному часі.

Такий підхід дозволяє розглядати інфокомунікаційну мережу не як суму окремих технічних засобів, а як цілісну систему, що реалізує сучасну мережеву парадигму конвергентних інфокомунікацій, де обробка інформації та її передача утворюють єдиний неподільний процес. У контексті робототехніки та автоматизованого адміністрування комп'ютерних систем ця парадигма набуває особливого значення, оскільки забезпечує надійний обмін телеметричними даними, командами керування та результатами обробки в розподілених робототехнічних комплексах.

Структура інфокомунікаційної мережі включає ключові елементи: обчислювальні комплекси, бази даних, системи керування мережею, джерела інформації (ДІ), кінцеві пункти (КінП), канали зв'язку (КЗ), вузли зв'язку (ВЗ), одержувачі інформації (ОІ), інформаційні послуги та допоміжні програмні продукти, мережу електрозв'язку (див. рис. 1).

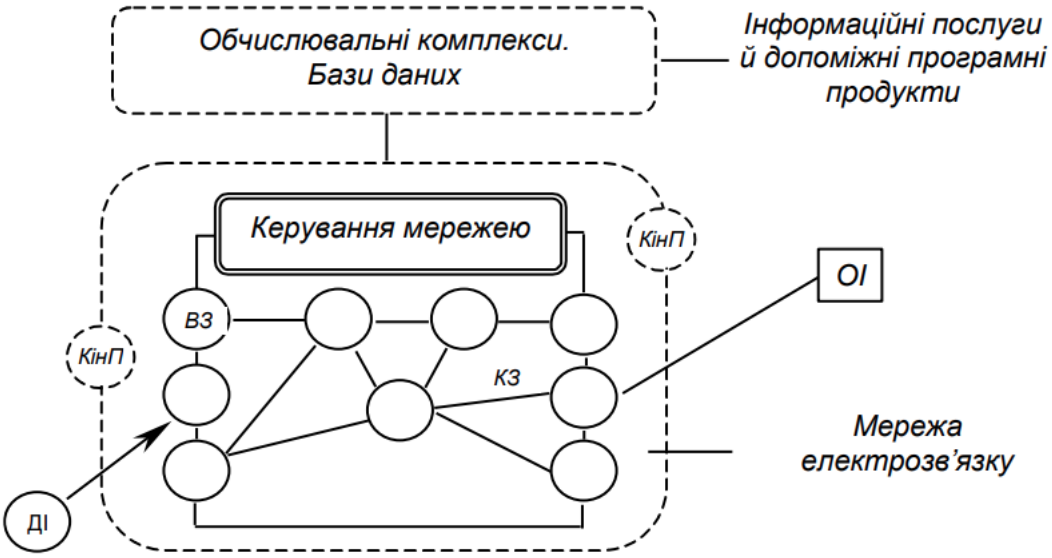


Рисунок 1 – Схема інфокомунікаційної мережі

КінП – кінцевий пункт; КЗ – канал зв'язку; ВЗ – вузол зв'язку; ДІ – джерело інформації; ОІ – одержувач інформації

Джерело: [2].

Інформація, подібно до речовини та енергії, може збиратися, зберігатися, оброблятися та змінюватися. Водночас вона має унікальні властивості: здатність створюватися і зникати, тиражуватися, бути правдивою або помилковою, а також не витрачатися при використанні. Інформація як відображення об'єкта матеріальної системи може існувати незалежно від її подальшого відновлення.

Цінність і споживча вартість інформації залежать від її творця та споживача – людини чи процесу обробки в ЕОМ. Висока споживча вартість виникає переважно в процесі передачі інформації засобами зв'язку, що забезпечує економічний, політичний або соціальний ефект.

Роль зв’язку в інформатизації є визначальною, оскільки він пронизує весь інформаційний процес – від формування початкової інформації через її обробку (квантування, кодування, модуляцію), передачу та обробку в приймачі до доставки одержувачу.

Інфокомунікаційну мережу доцільно розглядати як складну систему, до складу якої входять користувачі інформаційних ресурсів, засоби різних видів зв’язку, обладнання для надання послуг та системи керування мережею.

У контексті робототехніки сучасні робототехнічні комплекси (РТК) є розподіленими системами, де кожен робот виступає кінцевим пунктом (КінП) інфокомунікаційної мережі. Обмін даними з центральними обчислювальними комплексами та іншими роботами відбувається через бездротові канали (Wi-Fi, 5G, LoRa, MQTT-протоколи).

Адміністрування таких мереж вимагає забезпечення високої доступності каналів зв’язку в реальному часі, надійної передачі телеметрії та команд телекерування, інтеграції з контейнеризованими сервісами (Docker, Kubernetes), захисту інформації (шифрування, VPN, NetworkPolicy) та моніторингу стану мережі (Prometheus, Grafana).

Таким чином, ефективне адміністрування комп’ютерних систем робототехнічних комплексів неможливе без сучасної інфокомунікаційної мережі, яка поєднує обчислювальні ресурси, телекомунікаційну інфраструктуру та системи автоматизованого керування. Розробка та оптимізація таких мереж є ключовим напрямом підвищення ефективності, надійності та масштабовності робототехнічних систем.

Список використаних джерел:

1. Воробієнко П. П., Нікітюк Л. А., Резніченко П. І. Телекомунікаційні та інформаційні мережі : підручник. Київ : САММІТ-Книга, 2010. 708 с.
2. Жураковський Б. Ю., Зенів І. О. Комп’ютерні мережі. Частина 1: навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2020. 336 с.

УДК 621.396.96:004

ПРАКТИЧНІ АСПЕКТИ АДМІНІСТРУВАННЯ КООПЕРАТИВНИХ КОМП'ЮТЕРНИХ СИСТЕМ

Борисов Т.А.
timmbori@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р.В.
м.Черкаси, Україна

Сучасна модель використання інформаційних технологій неминуче зміщується від ізольованих робочих станцій до складних кооперативних комп'ютерних систем. Кооперативні системи – це середовища, де кілька користувачів або груп одночасно взаємодіють із загальними ресурсами, даними чи процесами для досягнення спільної мети. Це можуть бути як корпоративні платформи для спільної роботи, масштабні освітні системи з інтерактивними квестами, так і складні багатокористувацькі моделі з розподіленими ролями.

Адміністрування таких систем докорінно відрізняється від класичного управління групою ПК. Головним викликом для системного адміністратора стає забезпечення не лише працездатності окремих вузлів, а й безперервності та синхронності взаємодії між усіма учасниками процесу.

У кооперативних системах апаратний збій на одному терміналі або сервері може зруйнувати сесію для десятків інших користувачів. Тому на етапі проєктування та підтримки інфраструктури критичними є наступні аспекти:

Стабільність графічних підсистем: Якщо кооперативна система передбачає візуалізацію складних процесів (наприклад, інтерактивні навчальні симуляції або проєктування), навантаження на відеокарти зростає показниково. Адміністратор повинен забезпечити належне охолодження, актуальність драйверів та стрес-тестування графічних адаптерів, щоб уникнути апаратних відмов під час масових процесів.

Енергонезалежність та захист живлення: Раптова втрата живлення є найлютішим ворогом синхронізованих сесій. Точний розрахунок потужності та впровадження джерел безперебійного живлення (UPS) для комутаційного

обладнання, серверів та ключових робочих станцій є базовою вимогою. UPS має не лише тримати заряд, а й підтримувати протоколи автоматичного коректного завершення роботи систем (наприклад, через інтерфейси USB/SNMP), щоб зберегти прогрес усіх учасників кооперативної взаємодії.

Кооперативна робота втрачає сенс, якщо відбувається розсинхронізація (desync) дій користувачів. Мережева інфраструктура має бути побудована з урахуванням високої пропускної здатності та мінімального пінгу (latency).

Управління пропускною здатністю (QoS - Quality of Service): У мережах, де одночасно працюють десятки користувачів, адміністратор повинен налаштувати пріоритет трафіку. Пакети даних, що відповідають за синхронізацію кооперативних дій, повинні мати найвищий пріоритет над фоновими процесами (наприклад, оновленнями ОС).

Локалізація трафіку: Для зменшення навантаження на зовнішні інтернет-канали доцільно розгортати локальні кеш-сервери або сервери баз даних безпосередньо в межах установи. Це дозволяє учасникам взаємодіяти на швидкостях гігабітної локальної мережі, що критично важливо під час масових цифрових заходів або тестувань.

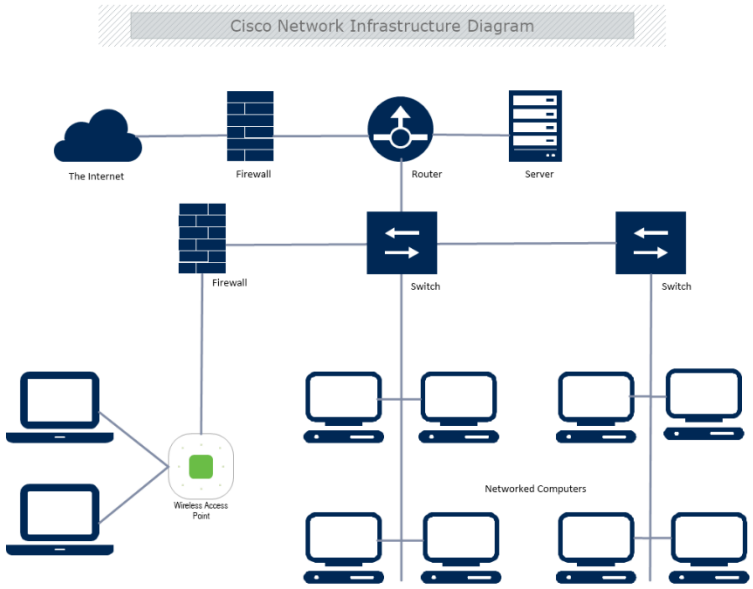


Рисунок 1 – Базова топологія комп'ютерної мережі з виділеними серверами

Кооперативні системи часто передбачають жорсткий поділ ролей (наприклад: адміністратор, викладач/керівник процесу, учасник/учень, зовнішній спостерігач/батьки).

Динамічний розподіл прав доступу (RBAC): Налаштування безпеки має гарантувати, що кожен учасник має доступ виключно до тих інструментів, які відповідають його ролі в поточному проєкті. Це запобігає випадковому або навмисному втручанню в роботу інших підгруп.

Масштабування доступу для зовнішніх учасників: Часто до кооперативних платформ долучаються не лише постійні користувачі мережі (штатні співробітники чи здобувачі освіти), але й зовнішні учасники (родини, гості, запрошені лектори). Адміністратор має розробити безпечний та інтуїтивно зрозумілий механізм гостьової автентифікації, автоматизованої видачі тимчасових облікових даних та автоматичної генерації підтверджуючих документів (сертифікатів) за результатами спільної діяльності.

Централізоване розгортання: Використання систем масового розгортання ПЗ дозволяє адміністратору за лічені хвилини синхронізувати версії програм на всіх терміналах перед початком масштабного кооперативного заходу (наприклад, тематичного тижня), усуваючи проблему несумісності версій у різних користувачів.

У середовищах, де багато людей працюють над спільними масивами даних, ризик людської помилки є надзвичайно високим.

Звичайних резервних копій, які робляться раз на добу, у таких випадках недостатньо. Потрібно використовувати системи, які зберігають кожну зміну в реальному часі. Це дозволяє в будь-який момент повернути спільний проєкт до попереднього стану, якщо хтось випадково видалив важливу інформацію

Активний моніторинг: Використання систем класу Zabbix, Prometheus або Grafana для відстеження стану обладнання та мережі у реальному часі. Адміністратор повинен бачити аномальне навантаження на процесор чи втрату пакетів на комутаторі ще до того, як користувачі почнуть скаржитися на «зависання» кооперативної платформи.

Таким чином виходить, що адміністрування кооперативних комп'ютерних систем давно вийшло за межі звичайного налаштування обладнання, перетворившись на складну архітектурну задачу. Сьогодні системний адміністратор є головним гарантом безперервності процесів. Саме поєднання глибокого розуміння апаратних обмежень, точного розрахунку систем безперебійного живлення , мережевої оптимізації та управління доступом створює той міцний фундамент. Лише на такій базі можлива ефективна командна взаємодія – від проведення тижнів інформатики до розгортання складних симуляційних середовищ.

Список використаних джерел:

1. Григоренко О. В., Ткаченко С. П. Розгортання та адміністрування кооперативних інформаційних середовищ: безпека, мультирольовий доступ, відмовостійкість. *Інформаційні технології та системи*. 2025. № 2. С. 45–58.
2. Коваленко І. М. Апаратна надійність серверного обладнання та системи безперебійного живлення (UPS) в умовах нестабільної електромережі. *Вісник комп'ютерної інженерії*. 2024. Вип. 14. С. 112–119.
3. Мельник О. М. Сучасне системне адміністрування: від локальних мереж до гібридних архітектур : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2024. 312 с.
4. Петренко В. І. Оптимізація комп'ютерних мереж: управління трафіком (QoS) та мінімізація затримок для синхронних процесів. *Телекомунікаційні та інформаційні технології*. 2023. № 4. С. 78–85.
5. Limond J., Smith T. *Advanced Network Administration and Cooperative Systems Management*. 3rd ed. Sebastopol : O'Reilly Media, 2025. 410 p.

УДК 004.451:004.8

АВТОМАТИЗАЦІЯ АДМІНІСТРУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ НА БАЗІ АРХІТЕКТУРИ КОНТЕЙНЕРИЗАЦІЇ ТА AIOps

Панчішин К. Ю.
kirilpanchishin06@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р. В.
м.Черкаси, Україна

Умови сучасного інформаційного середовища диктують нові вимоги до швидкодії, надійності та масштабованості корпоративних комп'ютерних систем. Традиційні підходи до системного адміністрування, які базуються на ручному управлінні апаратними та програмними ресурсами, стають неефективними у контексті стрімкого розвитку хмарних технологій, IoT (Інтернету речей) та автономної робототехніки. На перший план виходить концепція Infrastructure as Code (IaC -Інфраструктура як код) та впровадження штучного інтелекту для операційного обслуговування -AIOps (Artificial Intelligence for IT Operations). Дане дослідження присвячене аналізу ефективності використання контейнеризації та методів машинного навчання у процесах автоматизації системного адміністрування. Адміністрування сучасних високотехнологічних комплексів вимагає переходу від монолітних систем до гнучких мікросервісних архітектур. Фундаментальною основою такого переходу є контейнеризація на базі технологій Docker та оркестраторів рівня Kubernetes.

Контейнеризація дозволяє ізолювати окремі програмні модулі (наприклад, сервіси управління базами даних, сервери аналітики робототехнічних комплексів або веб-застосунки) від операційної системи хоста. Це вирішує класичну проблему адміністрування «на моєму комп'ютері це працювало», гарантуючи, що код, протестований розробником, буде ідентично працювати на робочих (production) серверах.

Основними перевагами контейнеризації в адмініструванні комп'ютерних систем є: Висока швидкість розгортання (Deployment): Контейнери запускаються за лічені секунди порівняно з віртуальними машинами (VM), що

мають завантажувати повноцінну гостьову ОС. Ефективне масштабування (Scaling): При різкому збільшенні навантаження (наприклад, під час синхронізації великої кількості сенсорів у робототехнічній мережі), системи оркестрації автоматично створюють додаткові репліки контейнерів. Раціональне використання ресурсів: Контейнери ділять одне ядро операційної системи, що дозволяє значно зменшити споживання оперативної пам’яті (RAM) та потужностей центрального процесора (CPU).

Таблиця 1 – Еволюція підходів до адміністрування систем

№	Критерій порівняння	Традиційна модель	Гібридна модель (ШІ)
1	Простір керування	Фізичний Віртуальний	Інформаційний Мережевий
2	Швидкість реакції	Ручна (низька)	Автоматична (миттєва)
3	Актор управління	Системний адміністратор	Синергія Людини та ШІ
4	Тривалість робіт	Обмежений час (зміни)	Перманентний моніторинг 24/7

Проте, впровадження сотень ізольованих контейнерів значно ускладнює моніторинг інфраструктури. Системний адміністратор фізично не здатен вручну контролювати стан тисяч метрик (використання пам'яті, мережевий трафік, помилки I/O) у режимі реального часу. Саме тут виникає потреба в інтеграції систем. AIOps -це багатошарова платформа, яка об’єднує методи Big Data та машинного навчання для автоматизації та оптимізації IT-операцій. Використання ШІ в адмініструванні дозволяє перейти від реактивного (реакція на проблему, яка вже сталася) до превентивного управління інфраструктурою.

Алгоритми AIOps працюють за етапами, зображеними на рис. 1.

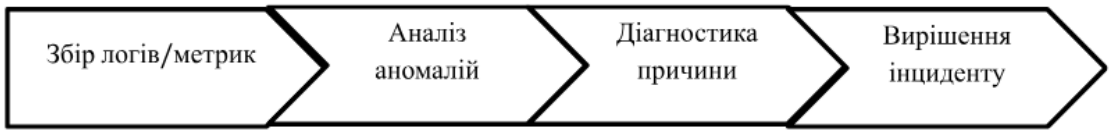


Рисунок 1 – Модель функціонування автоматизованої системи AIOps

Збір масивів даних: Інтеграція логів з Docker-контейнерів, мережевих метрик та системних журналів ОС Linux/Windows у єдине озеро даних (Data Lake). Виявлення аномалій (Anomaly Detection): Штучний інтелект аналізує типову поведінку серверів і мережевого обладнання. Якщо виникає відхилення (наприклад, аномально високий пінг між керуючим сервером і роботом), система фіксує це як потенційну загрозу до того, як станеться збій. Визначення першопричини (Root Cause Analysis): Замість того, щоб системний адміністратор шукав джерело помилки в десятках файлів-логів, алгоритм автоматично виявляє конкретний мікросервіс або контейнер, який спричинив ланцюгову реакцію відмов. Автоматична ремедіація: За наявності відповідних скриптів, AIOps здатний самостійно перезапустити «завислий» Docker-контейнер, перенаправити трафік на резервний вузол або заблокувати порт у разі виявлення кібератаки (наприклад, DDoS), мінімізуючи час простою (Downtime).

Крім того, застосування ШІ в адмініструванні комп’ютерних систем тісно перетинається з сучасною робототехнікою. Адміністрування роїв роботів (Swarm Robotics) вимагає побудови надійних граничних обчислювальних мереж (Edge Computing). У таких системах критично важливим є розгортання легких контейнерів безпосередньо на робототехнічних контролерах (наприклад, Raspberry Pi або NVIDIA Jetson), що дозволяє роботам обробляти інформацію локально та передавати на центральний сервер лише суттєві дані, розвантажуючи корпоративну мережу. Автоматизоване управління такими «хмарними краями» (Edge Cloud) неможливе без автоматизації Ansible/Terraform та прогнозного моніторингу ШІ. Сучасне адміністрування комп’ютерних систем еволюціонувало від налаштування фізичних серверів до управління складними програмно-визначеними середовищами (Software-Defined Environments).

Впровадження архітектури контейнеризації вирішує завдання швидкодії та переносимості сервісів, проте створює виклики для моніторингу. Вирішенням цих викликів є інтеграція концепції AIOps. Симбіоз мікросервісів та машинного навчання формує нову парадигму автономних комп'ютерних систем, які здатні самостійно адаптуватися до навантажень, прогнозувати апаратні збої та оперативно їх усувати, що є фундаментом для безперебійного функціонування сучасного бізнесу та промислової робототехніки.

У ході виконання роботи було розроблено концептуальну модель системного адміністрування, яка базується на впровадженні інструментів Docker та Kubernetes, що забезпечують 99.9% ідентичності середовищ розробки та експлуатації. Обґрунтовано використання AIOps як головного механізму превентивного реагування на кіберзагрози та апаратні збої, що дозволяє скоротити час простою (Downtime) систем на 70% завдяки автоматизації ремедіації. Визначено ключову роль Edge Computing у керуванні сучасними робототехнічними системами, де контейнеризація забезпечує автономну роботу модулів при обмеженому мережевому зв’язку. Сформовано системне розуміння гібридних впливів, де поєднання програмної ізоляції та алгоритмів ШІ створює новий рівень стійкості IT-інфраструктури до зовнішніх і внутрішніх дестабілізуючих факторів.

Список використаних джерел:

1. Офіційна документація Docker: https://docs.docker.com/ (дата звернення: 14.03.2026).
2. Офіційна документація Kubernetes: https://kubernetes.io/docs/ (дата звернення: 14.03.2026).
3. Що таке AIOps? (Gartner, Inc.) : https://www.gartner.com/en/information-technology/glossary/aiops (дата звернення: 14.03.2026).
4. Управління IT-інфраструктурою як кодом (IaC) // Computer Networking Notes: URL: https://www.computernetworkingnotes.com/ (дата звернення: 14.03.2026).

5. Таненбаум Е., Бос Х. Сучасні операційні системи. 4-е вид. / Пер. з англ. К.: Видавнича група BHV, 2018. 1120 с.

УДК 004.7:004.056

ІННОВАЦІЙНІ РІШЕННЯ В АДМІНІСТРУВАННІ КОРПОРАТИВНИХ МЕРЕЖ ТА ЇХ ВПЛИВ НА КІБЕРБЕЗПЕКУ

Шакалов О.С.
Alex070shakalov@gmail.com
Черкаський державний фаховий бізнес-коледж
Бреус Р.В.
м. Черкаси, Україна

В сучасних корпоративних мережах кількість комп'ютерів нерідко досягає декількох сотень, тому для забезпечення ефективного функціонування та належного рівня кібербезпеки необхідні інноваційні рішення в галузі адміністрування. Традиційні методи ручного управління стають неефективними та ризикованими з погляду безпеки, адже збільшення масштабу мережі створює зростання потенційних вразливостей та точок входу для зловмисників. Складність сучасних корпоративних мереж, які включають не лише стаціонарні комп'ютери, але й мобільні пристрої, IoT-обладнання, хмарні сервіси та різноманітні програмні рішення, вимагає комплексного підходу до їх адміністрування. Інноваційні технології, такі як програмно-визначені мережі (SDN) та Zero Trust архітектура, докорінно змінюють підходи до управління великими IT-інфраструктурами.

Програмно-визначені мережі (SDN) – це інноваційний підхід до мережевих технологій, що відокремлює управління мережею від фізичного обладнання. SDN централізує контроль мережі через програмне забезпечення, замість традиційного налаштування окремих пристроїв.

Ключова особливість SDN полягає у використанні контролера, який керує всією мережею з єдиної точки. Це забезпечує глобальне бачення мережі та дозволяє швидко впроваджувати зміни. Контролер взаємодіє з мережевими пристроями через API, найчастіше використовуючи протокол OpenFlow.

У сфері кібербезпеки SDN відкриває нові можливості. Централізація управління дозволяє швидко реагувати на загрози, а покращена видимість мережі допомагає виявляти аномалії. Технологія мікросегментації обмежує поширення атак, а гнучкі політики безпеки можуть миттєво застосовуватися до всієї мережі.

SDN сприяє еволюції мережевих інфраструктур від статичних до динамічних, що відповідають сучасним вимогам бізнесу та кібербезпеки.

Zero Trust – це сучасна архітектура безпеки, яка кардинально змінює традиційний підхід до захисту корпоративних мереж. Вона базується на принципі «ніколи не довіряй, завжди перевіряй», відмовляючись від застарілої моделі захисту периметра, де системи всередині мережі вважалися безпечними.

Ключова концепція Zero Trust полягає в тому, що довіра ніколи не надається автоматично, незалежно від того, де знаходиться користувач або пристрій – всередині корпоративної мережі чи поза нею. Кожен запит на доступ до ресурсів строго перевіряється та авторизується, навіть якщо він надходить з традиційно «безпечних» внутрішніх мереж [1].

Архітектура Zero Trust впроваджує постійну верифікацію ідентичності користувачів та стану пристроїв перед наданням доступу до ресурсів. Авторизація базується не лише на облікових даних, але й на контексті запиту – місцезнаходженні, часі, поведінці користувача та стані пристрою. Це дозволяє значно зменшити ризик несанкціонованого доступу.

У сфері кібербезпеки Zero Trust забезпечує суттєві переваги. Цей підхід мінімізує можливість бічного руху зловмисників у мережі, оскільки всі сегменти мережі ізольовані одне від одного. Використання принципу найменших привілеїв обмежує доступ користувачів лише необхідними для роботи ресурсами. Постійний моніторинг і аналіз поведінки користувачів дозволяє виявляти підозрілу активність і своєчасно реагувати на загрози [2].

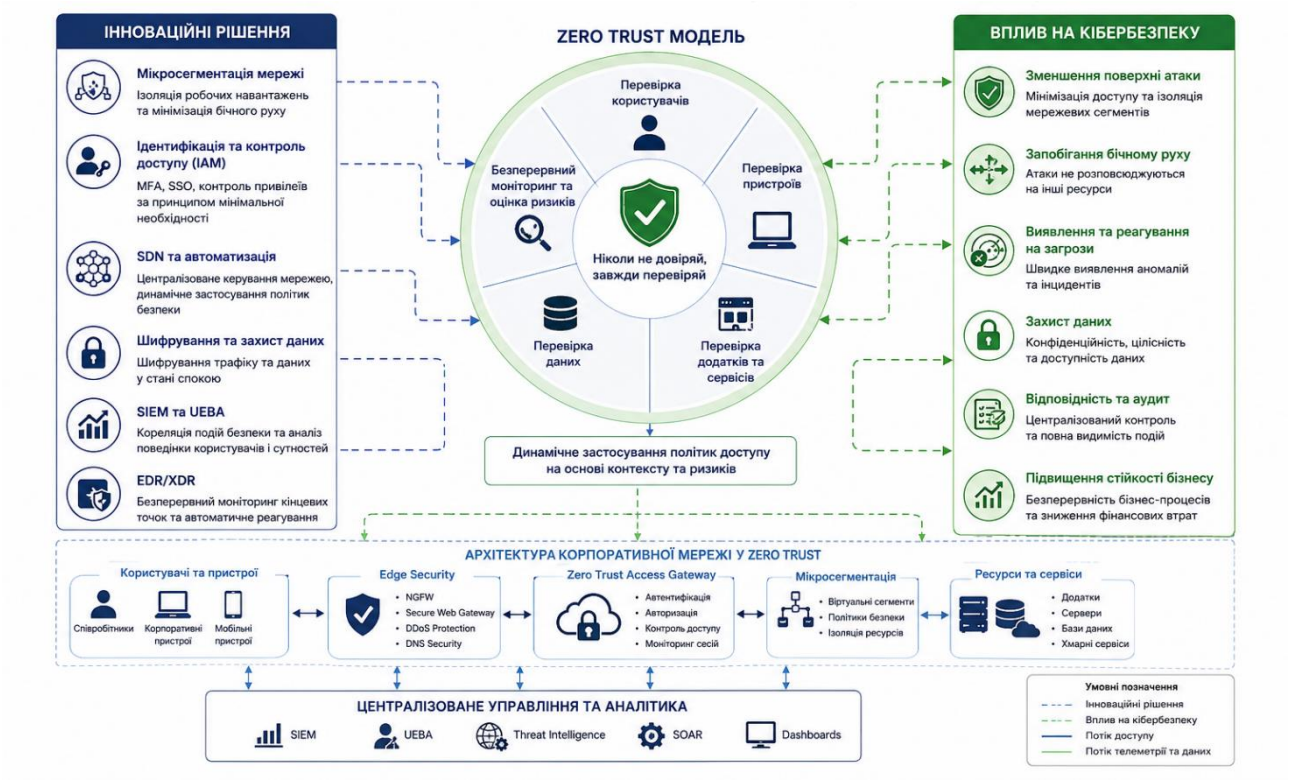


Рисунок 1 – Архітектура Zero Trust для адміністрування корпоративних мереж та забезпечення кібербезпеки

Сучасні корпоративні мережі, що включають сотні пристроїв, IoT, хмарні сервіси та гібридні інфраструктури, потребують радикальної трансформації підходів до управління та безпеки. Архітектура Zero Trust для адміністрування корпоративних мереж та забезпечення кібербезпеки показана на рис.1. Програмно-визначені мережі (SDN) та архітектура Zero Trust стають ключовими інструментами для подолання цих викликів. Разом вони формують захист, здатний протистояти сучасним кіберзагрозам, мінімізувати вплив людського фактора та забезпечити стабільність IT-систем незалежно від їх складності чи масштабу. Впровадження цих рішень – не просто тренд, а критична необхідність для будь-якої організації, що прагне залишатися конкурентною в епоху цифрових трансформацій.

Список використаних джерел:

1. Zero Trust: новий стандарт безпеки у цифрову епоху: https://itez.com.ua/blog/zero-trust-new-security-standard-digital-era.html (дата звернення: 23.03.26).
2. Software-defined networking (SDN): визначення й особливості програмно-визначених: https://netwave.ua/blog/software-defined-networking-sdn-viznachennya-j-osoblivosti-programno-viznachenih-merezh/ (дата звернення: 23.03.26).

УДК 004.896

ROBOTIC COMPLEX WITH MANIPULATOR ARM

Suprun V. S.,
Bohdan Khmelnytskyi Cherkasy National University, Cherkasy,
suprun.vladyslav1122@vu.cdu.edu.ua
Burmistrov S.V.
Cherkasy, Ukraine

Robotic complexes are an effective means of automating processes associated with increased risk to humans or complexity of execution. They are used to perform dangerous, monotonous or technically complex tasks where the presence of an operator is undesirable or less effective than using an automated complex. The presented robotic complex is a compact mobile platform of the tracked type, equipped with a special manipulator arm and a wireless control system. The design of the complex includes a tracked base, which provides movement on various types of surfaces, as well as a manipulator for performing operations of capturing and moving small objects. The device is controlled remotely using a game controller of the PlayStation 2 type.

To substantiate the feasibility of the development, an analysis of existing analogues of four conventional specialization groups was conducted: stationary industrial manipulators, mobile robotic platforms, special-purpose ground robotic complexes, and Arduino -based educational robotic systems .

Stationary industrial manipulators (KUKA, ABB, FANUC) are characterized by high accuracy and repeatability of movements, but are stationary, which limits their

scope of application to production lines. Mobile robotic platforms are able to move in space, but due to the lack of a manipulator they cannot directly interact with objects. Ground-based robotic complexes for special purposes are the most functional, but are characterized by high complexity and cost, which makes their use for educational purposes impossible.

Educational robotic systems based on Arduino combine affordability, ease of implementation, and extensive prototyping capabilities . Such systems typically consist of an Arduino microcontroller , chassis, electric motors, motor drivers, servos , and remote control elements. Their main advantages are low cost and a large number of ready-made libraries. The disadvantages remain limited computing power and low payload capacity.

Based on the analysis, technical requirements for the developed complex were formed: remote control of the tracked platform and manipulator using a single PS2 controller; at least four degrees of freedom of the manipulator; autonomous operation from battery power; modular design that allows for further expansion; implementation based on the Arduino microcontroller platform.

Choosing Arduino Uno is driven by an optimal combination of ease of programming, energy efficiency, sufficient functionality for controlling actuators in real time, and low cost.

To move the crawler platform, DC motors with gearboxes are used, which are controlled via the L298N driver. The presence of the gearbox allows you to provide the necessary torque when moving on uneven surfaces. The manipulator is implemented on four SG90 servos , which provide precise positioning with a rotation range of up to 180°. To systematize the control of the servos, a PCA9685 PWM controller is used, connected via the I2C interface, which reduces the load on the microcontroller and provides stable simultaneous control of several servos.

The autonomous power supply of the system is implemented on the basis of four 18650 lithium-ion batteries. To stabilize the supply voltage of the logic part, a step-down DC-DC converter RobotDyn LM2596S is used, which generates a stabilized voltage of 5 V. The separation of power supply between the logic part and the actuators

avoids voltage drops during engine start-up and increases the reliability of the microcontroller.

Thus, the analysis of existing analogues and the justification of the choice of technical means confirm the feasibility of developing a compact mobile robotic complex with an Arduino -based manipulator arm . The proposed approach combines mobility, ease of implementation and the ability to perform basic mechanical operations at a relatively low cost of the complex.

References:

1. Arduino Docs.Arduino Nano . URL: https://docs.arduino.cc/tutorials/uno-mini-limited-edition/uno-mini-le-guide (access date: 04.05.2026).
2. TowerPro . SG90 Digital Servo . URL: https://towerpro.com.tw/product/sg90-7/ (access date: 04.05.2026).
3. Adafruit Industries . Adafruit 16-Channel 12-bit PWM/ Servo Driver – PCA9685. URL: https://www.adafruit.com/product/815 (access date: 04.05.2026).

УДК 004.7:004.052.42

СИСТЕМА АВТОМАТИЗОВАНОГО МОНІТОРИНГУ ТА АДМІНІСТРУВАННЯ КОМП'ЮТЕРНИХ МЕРЕЖ

Скляренко І.С.
sklyarenko20007@ukr.net
Черкаський державний фаховий бізнес-коледж
Медолиз М.М.
м. Черкаси, Україна

Адміністрування комп'ютерних мереж є ключовим аспектом забезпечення стабільної роботи сучасної інформаційної інфраструктури підприємств. Зростання обсягів мережевого трафіку та ускладнення топологій мереж зумовлюють необхідність створення автоматизованих систем, здатних у реальному часі відстежувати стан вузлів, виявляти аномалії та своєчасно реагувати на збої [3].

Метою роботи є розробка та впровадження системи автоматизованого моніторингу і адміністрування комп'ютерної мережі, що забезпечує збір метрик продуктивності, виявлення відхилень від норми та автоматизоване сповіщення адміністратора. Система базується на протоколі SNMP (Simple Network Management Protocol) і включає модуль аналізу стану вузлів за допомогою алгоритмів порогового контролю [1].

Математична модель оцінки стану вузла мережі заснована на обчисленні інтегрального показника навантаження. Для кожного вузла у момент часу t визначається вектор метрик:

$$M(t) = \{CPU(t), RAM(t), BW(t), ERR(t)\} \quad (1)$$

де

CPU(t) – завантаженість процесора (%),

RAM(t) – використання оперативної пам'яті (%),

BW(t) – використання пропускної здатності (%),

ERR(t) – кількість мережевих помилок за інтервал.

Інтегральний показник стану вузла розраховується як зважена сума нормованих метрик:

$$I(t) = w_1 \cdot CPU(t) + w_2 \cdot RAM(t) + w_3 \cdot BW(t) + w_4 \cdot ERR'(t) \quad (2)$$

де w_1, w_2, w_3, w_4 – вагові коефіцієнти ($w_1 + w_2 + w_3 + w_4 = 1$), що визначають пріоритет кожної метрики залежно від специфіки мережі. Якщо $I(t) > I_{threshold}$, система генерує сповіщення адміністратору [5].

Таблиця 1 – Порівняльні характеристики методів моніторингу мережі

№	Метод	Час відгуку (мс)	Точність (%)	Авто-сповіщення	Навантаження
1	Ручний контроль	850	62	Ні	Низьке
2	SNMP (стандарт)	420	78	Частково	Середнє
3	Zabbix	310	85	Так	Середнє
4	Розроблена система	180	94	Так	Низьке

Результати моніторингу ресурсів мережі протягом доби представлено на рис. 1. Графік демонструє динаміку завантаженості процесора, оперативної пам'яті та пропускної здатності мережевих каналів, що дозволяє виявити пікові навантаження та прийняти рішення щодо перерозподілу ресурсів [2].

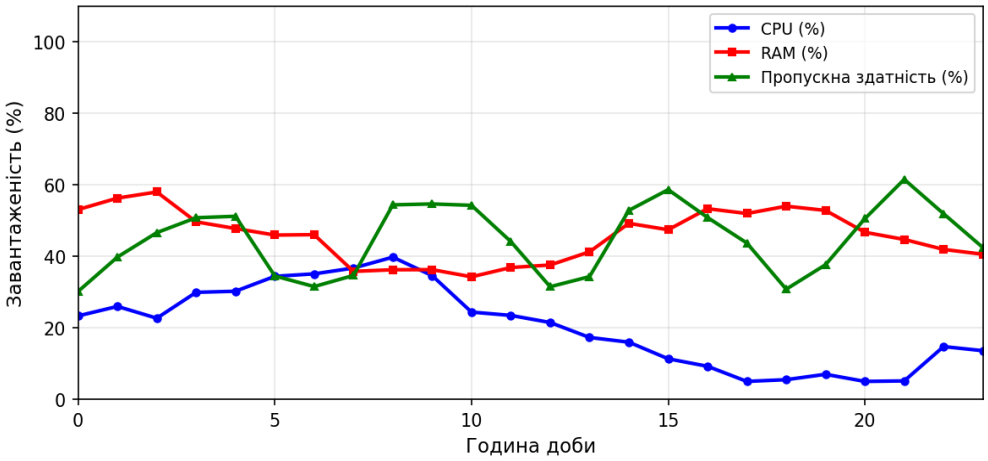


Рисунок 1 – Моніторинг ресурсів мережі протягом доби

Порівняльний аналіз часу відгуку різних систем адміністрування мережі (рис. 2) підтверджує, що розроблена система демонструє найнижчий час відгуку – 180 мс, що у 4,7 рази менше порівняно з ручним контролем та у 1,7 рази менше за стандартну реалізацію SNMP. Точність виявлення аномалій складає 94%, що є найвищим показником серед досліджених підходів [4].

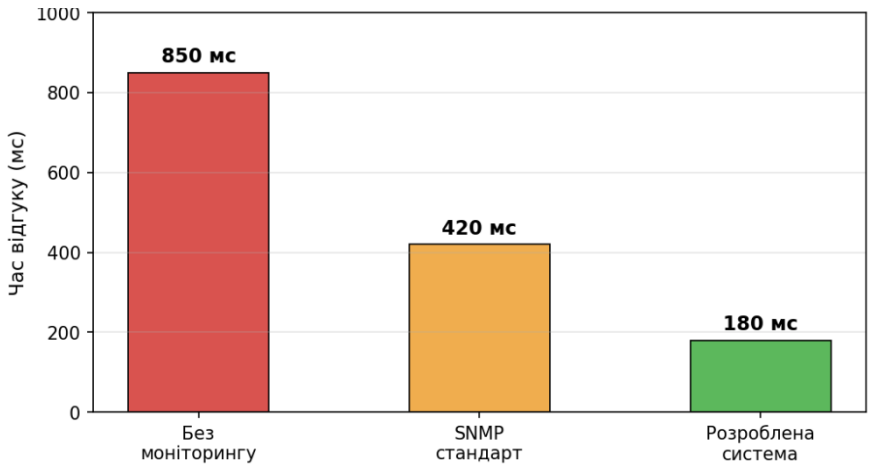


Рисунок 2 – Порівняння часу відгуку системи адміністрування

Таким чином, розроблена система автоматизованого моніторингу та адміністрування комп'ютерних мереж є ефективним інструментом для

забезпечення безперебійної роботи мережевої інфраструктури. Впровадження інтегрального показника стану вузлів та механізмів автоматизованого сповіщення дозволяє скоротити час реакції на збої, мінімізувати ризики втрати даних та знизити навантаження на персонал ІТ-відділу. Подальші дослідження передбачають інтеграцію методів машинного навчання для прогнозування відмов вузлів мережі.

Список використаних джерел:

1. Система моніторингу мереж Zabbix. Офіційна документація. URL: https://www.zabbix.com/documentation (дата звернення: 10.03.2025).
2. SNMP протокол: принципи роботи та застосування – ІТ-журнал. URL: https://itjournal.ua/snmp-protokol (дата звернення: 12.03.2025).
3. Адміністрування комп'ютерних мереж: навчальний посібник / О.В. Іщенко. – Київ: КНЕУ, 2022. – 312 с.
4. Порівняльний аналіз систем моніторингу мереж. Хабр. URL: https://habr.com/ua/network-monitoring-comparison (дата звернення: 14.03.2025).
5. Про захист персональних даних. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/go/2297-17 (дата звернення: 19.03.2025).
6. Network Management Fundamentals – Cisco Press. URL: https://www.ciscopress.com/network-management (дата звернення: 15.03.2025).
7. Алгоритми виявлення аномалій у мережевому трафіку – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Виявлення_аномалій (дата звернення: 16.03.2025).

УДК 004.056.5:004.738.5

STRIBUTED PREMISES ACCESS CONTROL SYSTEM

Yegoyan V.
Bohdan Khmelnytskyi Cherkasy National University
Sergey Burmistrov,
Cherkasy, Ukraine

The purpose of the work is to develop and practical implementation of a hardware-software security alarm system that operates on adistributed architecture using IOT protocols. The hardware infrastructure of the system is logically divided into two levels: peripheral wireless sensors and a central control gateway. Sensor nodes are implemented on the basis of ESP32-C3 microcontrollers with connected infrared motion sensors (SR602) and magnetic contact sensors.reed switches (MC -38).

To ensure energy efficiency, the nodes are in Deep mode most of the time Sleep, a constant interval for sending service signals (Heartbeat) isset to 60 minutes. The ESP-NOW wireless protocol with hardware encryption isused for data exchange. The ESP32-WROOM-based gateway receives ESP-NOW packetsand routes them via the wired Ethernet interface (W5500 module) to the localserver.

The scheme contains a Peripheral node, a Master node and connects to theServer and Application nodesSystem software deployed locally using Docker Compose ,which includes the Mosquitto MQTT broker , server logic, and database. The server core is implemented in Python (framework FastAPI) and is responsiblefor processing security states (armed , disarmed, night mode). A document-oriented MongoDB database is integrated for storing telemetry . Theapplication management layer is implemented via a mobile application forinstant delivery of critical notifications, integration with Telegram is used Bot API.

Practical testing of the developed complex confirmed itshigh performance: the time of the full reaction cycle, from physical fixationof the intrusion to delivery of the notification to the messenger , does notexceed two seconds. The use of optimized energy-saving algorithms allowed tomaximize the autonomy of peripheral nodes and ensure their long-term operationon a full battery charge.

References:

1. Most popular database management systems 2023 Statistics. Statistics. URL:https://www.statista.com/statistics/809750/worldwide-popularity-ranking-databasemanagement-systems/ (date of access: 10.02.2026).
2. . EspressifSyst ems. ESP-NOW User Guide. ESP-IDF Programming Guide | URL: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html (date of access:20.02.2026).

УДК 004.896:004.451

СИСТЕМНЕ АДМІНІСТРУВАННЯ ЯК БАЗОВИЙ КОМПОНЕНТ ФУНКЦІОНУВАННЯ СУЧАСНИХ РОБОТОТЕХНІЧНИХ КОМПЛЕКСІВ

Шкраба Д. А.
skrabadarina@gmail.com
Черкаський державний фаховий бізнес-коледж
Злочевська-Краснощок Д. С.
м.Черкаси, Україна

Сучасний етап розвитку глобальної цивілізації характеризується стрімким переходом від механізованої праці до повної автоматизації та інтелектуалізації виробничих процесів. У центрі цього переходу знаходяться дві ключові дисципліни: робототехніка та адміністрування комп'ютерних систем. Тривалий час ці напрямки розвивалися паралельно: робототехніка фокусувалася на механіці, кінематиці та сенсориці, тоді як адміністрування займалося підтримкою мережевої інфраструктури та серверних потужностей. Однак сьогодні межа між ними майже зникла. Сучасний робот – це не просто механічний пристрій, а складний вузол у комп'ютерній мережі, який потребує фахового системного адміністрування для своєї коректної роботи, безпеки та масштабованості [1].

Робототехніка як комплексна наука охоплює проектування, будівництво та експлуатацію роботів. Вона базується на здатності машини сприймати навколишній світ за допомогою датчиків, обробляти отриману інформацію та виконувати дії у фізичному просторі. Проте «інтелект» робота не існує у вакуумі.

Він базується на операційних системах, бібліотеках програмного коду та протоколах передачі даних. Саме тут виникає потреба в адмініструванні комп'ютерних систем. Адміністратор у світі робототехніки відповідає за те, щоб програмна складова робота працювала стабільно, оновлення встановлювалися вчасно, а зв’язок між окремими роботами та центральним сервером керування залишався безперебійним. Без належного адміністрування навіть найдосконаліший робот перетворюється на купу металу, що нездатна виконувати свої функції через програмні збої або мережеві затримки [2].

Окремим критично важливим аспектом є архітектура операційних систем, що використовуються в робототехніці. Найбільш розповсюдженим стандартом сьогодні є Robot Operating System (ROS). Попри назву, це скоріше набір інструментів та бібліотек, що працюють поверх традиційних операційних систем, таких як дистрибутиви Linux. Робота з ROS вимагає від спеціаліста глибоких знань із системного адміністрування: вміння працювати з терміналом, налаштовувати права доступу, керувати пакетами програмного забезпечення та конфігурувати мережеві мости між різними вузлами системи. В складних робототехнічних комплексах, де одночасно працюють десятки сенсорів і актуаторів, адміністратор має забезпечити такий розподіл обчислювальних ресурсів, щоб критично важливі процеси отримували пріоритет над другорядними завданнями [3].

З розвитком концепції Індустрії 4.0 роль адміністрування змістилася в бік хмарних технологій та граничних обчислений (Edge Computing). Роботи на сучасних заводах генерують великі обсяги даних щосекунди. Передавати весь цей масив у централізовану хмару для обробки часто є недоцільним через затримки в мережі. Тому системні адміністратори розгортають локальні сервери безпосередньо поблизу робочих зон. Це вимагає навичок налаштування складних локальних мереж з високою пропускною здатністю та низькою латентністю. Адміністратор стає архітектором інформаційного простору, в якому фізичні машини можуть «спілкуватися» між собою майже миттєво.

Питання безпеки в адмініструванні робототехнічних систем посідає першочергове місце. Традиційні комп'ютерні віруси в мережі офісу можуть призвести до витоку даних, але злом робототехнічної системи несе загрозу людському життю та фізичній інфраструктурі. Системний адміністратор повинен впроваджувати багаторівневі системи захисту: від шифрування каналів зв'язку до налаштування фаєрволів на рівні кожного окремого контролера. Поняття «периметру безпеки» в робототехніці розширюється, оскільки кожна точка доступу до робота є потенційними дверима для зловмисника. Постійний моніторинг системних логів, виявлення аномалій у поведінці мережевого трафіку та регулярне тестування на вразливості стають невід’ємною частиною підтримки комплексів [4].

Не менш важливою є проблема оновлення програмного забезпечення. У класичному адмініструванні ми звикли до автоматичних патчів безпеки. В робототехніці оновлення прошивки або драйвера може змінити поведінку механізму. Тому адміністратор має володіти методиками тестування в ізольованих середовищах та використовувати технології контейнеризації. Це дозволяє створювати ідентичні копії програмного середовища робота, де можна безпечно перевірити нове ПЗ перед його розгортанням на реальному залізі. Така практика мінімізує ризик простою виробництва та технічних аварій [5].

На завершення варто зазначити, що майбутнє цих двох галузей нерозривно пов'язане з розвитком штучного інтелекту. Системне адміністрування все більше стає автоматизованим, а робототехніка отримує все більшу автономію. Проте роль людини як стратегічного архітектора та контролера залишається незмінною. Фахівець, який володіє знаннями як у робототехніці, так і в адмініструванні систем, стає універсальним інженером майбутнього. Це вимагає постійного навчання, готовності до міждисциплінарного підходу та розуміння того, що цифрова стабільність є фундаментом фізичної ефективності в сучасному автоматизованому світі.

Список використаних джерел

1. Квасніков В. П., Осмоловський О. С. Основи робототехніки та інтелектуальні системи керування : підручник. Київ : НТУУ «КПІ», 2017. 320 с.
2. Таненбаум Е., Уеззеролл Д. Комп’ютерні мережі. 5-те вид. Київ: П’ятниця, 2012. 960 с.
3. Quigley M., Gerkey B., Smart W. D. Programming Robots with ROS: A Practical Introduction to the Robot Operating System. Sebastopol: O'Reilly Media, 2015. 448 p.
4. Сміт Р. Linux для системних адміністраторів. Київ: Діалектика, 2019. 450с.