МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
НАУКОВА АСОЦІАЦІЯ КІБЕРБЕЗПЕКИ УКРАЇНИ

УДК 004.4

Тези доповідей

IX Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні технології"

23 квітня 2026 року

Кропивницький 2026

УДК 004.4

Матеріали IX Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології": тези доповідей, 23 квітня 2026 р. – Кропивницький: ЦНТУ, 2026. – 109 с.

Наведені тези пленарних та секційних доповідей за теоретичними та практичними результатами наукових досліджень і розробок. Представлені результати теоретичних досліджень в галузях проектування інформаційних систем, технологій захисту інформації, використання сучасних інформаційних технологій в управлінні системами за різними галузями народного господарства.

Матеріали публікуються в авторській редакції.

За достовірність викладених фактів, цитат та інших відомостей відповідальність несуть автори.

СЕКЦІЯ 1. ІНФОРМАЦІЙНА БЕЗПЕКА ДЕРЖАВИ, СУСПІЛЬСТВА ТА ОСОБИСТОСТІ

УДК 004.056:004.8:005.52

Є.В. Пастух[1], *здобувач першого курсу першого рівня вищої освіти*
М.В. Козак[1], *здобувач першого курсу першого рівня вищої освіти*
Н.С. Петляк[1], *доктор філософії, доцент кафедри кібербезпеки*
pastukhyv@khmnu.edu.ua, kozakmv@khmnu.edu.ua, npetlyak@khmnu.edu.ua
[1]Хмельницький національний університет, Хмельницький

ШТУЧНИЙ ІНТЕЛЕКТ ЯК ІНСТРУМЕНТ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ АУДИТУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

У сучасних умовах стрімкої еволюції кіберзагроз, традиційні методи аудиту систем захисту інформації дедалі частіше виявляються недостатньо ефективними. З огляду на складність і динамічність інформаційного середовища, виявлення прихованих аномалій та латентних вразливостей за допомогою звичайних підходів ускладнюється, адже навіть досвідчені аудитори не завжди здатні ідентифікувати мінімальні відхилення у поведінці систем безпеки. У цьому контексті особливого значення набуває впровадження технологій штучного інтелекту (ШІ), зокрема штучних нейронних мереж і методів машинного навчання, які можуть значно підвищити ефективність аудиту інформаційної безпеки (ІБ). Застосування інструментів ШІ у процесі аудиту дозволяє автоматизувати рутинні операції, зменшити вплив людського фактора, а також забезпечити обробку великих обсягів даних, що надходять з розгалужених і складних інфраструктур. Завдяки алгоритмам навчання з підкріпленням, кластеризації та детекції аномалій, стає можливим глибокий аналіз поведінкових шаблонів користувачів і систем, що особливо актуально при проведенні як зовнішнього, так і внутрішнього аудиту.

Технології ШІ набувають дедалі більшого поширення у різних сферах суспільного життя, стаючи фундаментом для нової хвилі цифрової трансформації. Сьогодні провідні транснаціональні корпорації активно впроваджують інструменти штучного інтелекту в комерційні продукти та операційні процеси. Так, компанія Microsoft інтегрувала інтелектуального асистента Copilot на базі нейромереж і архітектури GPT-4 безпосередньо в операційну систему Windows 11, що є яскравим прикладом переходу від експериментального використання ШІ до повсякденної функціональності. У цьому контексті надзвичайно актуальним стає питання про доцільність застосування ШІ у професійній сфері, зокрема в аудиті ІБ. ШІ уже продемонстрував значний потенціал у галузі фінансового аудиту. Його здатність до обробки великих обсягів даних у поєднанні з машинним навчанням дозволяє ефективно виявляти аномальні транзакції, фінансові невідповідності та спроби маніпуляцій з балансом. Крім того, технології обробки природної мови (Natural Language Processing, NLP) успішно застосовуються для автоматизації рутинних процедур аналізу нормативної та внутрішньої документації, складання аудиторських висновків, генерації звітів. Це суттєво підвищує продуктивність аудиторських команд і зменшує вплив людського чинника.

Попри очевидні переваги, повноцінне впровадження ШІ в аудит ІБ наразі стримується низкою об'єктивних і суб'єктивних чинників. Серед основних причин, які спонукають аудиторські групи утримуватись від використання ШІ, варто виокремити відсутність належної професійної підготовки та технічної інфраструктури. Значна частина фахівців просто не володіє достатніми знаннями про функціонування інтелектуальних систем або не має доступу до сучасного програмного забезпечення. Водночас серед практиків поширене уявлення про надмірну вартість впровадження таких технологій, що робить їх економічно недоцільними в межах обмежених бюджетів окремих аудитів. Іншою поширеною причиною є переконання у недоцільності використання ШІ в аудиті, що часто ґрунтується на хибному уявленні про автоматизацію як загрозу професійній експертизі. Крім того, низка аудиторських компаній стикається з обмеженим доступом до конфіденційної або критично важливої інформації, що унеможливлює повноцінне навчання моделей та аналіз внутрішніх систем. До переліку бар'єрів можна також віднести складність у використанні інструментів ШІ, слабкий контроль за цілісністю оброблюваних даних, побоювання щодо порушення конфіденційності, відсутність усталених стандартів для інтеграції інтелектуальних систем у процес аудиту та труднощі з інтерпретацією результатів, отриманих унаслідок автоматизованої аналітики.

У сукупності ці фактори формують бар'єри, які потребують комплексного подолання. Одним із напрямів вирішення проблеми є формування методології використання ШІ в аудиті з чітким визначенням допустимих меж автоматизації, сценаріїв використання, вимог до вхідних даних і критеріїв оцінки якості результатів. У рамках цього підходу ШІ має розглядатися не як загроза для аудиторської професії, а як інструмент, що доповнює людську експертизу, розширює її можливості та дозволяє зосередитися на стратегічних і комплексних аспектах перевірки.

Перед застосуванням інструментів ШІ у сфері аудиту ІБ необхідно осмислити структуру самого аудиторського процесу. Аудит ІБ охоплює дві основні складові: управління програмою аудиту та безпосереднє

проведення аудиту. Методологічно структура аудиту узгоджується з циклом Демінга (PDCA — Plan, Do, Check, Act) (рис.1), що забезпечує безперервність удосконалення системи управління ІБ.

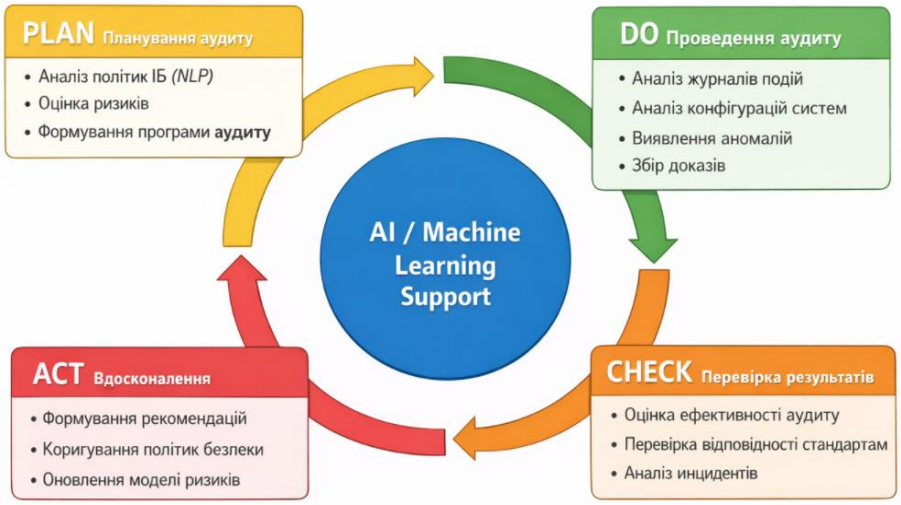


Рис. 1. Модель інтеграції ШІ в цикл аудиту ІБ

На етапі планування у межах менеджменту програми аудиту визначаються цілі, здійснюється оцінка ризиків та можливостей, розробляється програма аудиту. У власне процедурі аудиту планування включає ініціалізацію аудиту та підготовку до його проведення. Етап "Do" охоплює реалізацію програми та безпосереднє проведення аудиту з формуванням звіту. Етап "Check" — це оцінка ефективності аудиту, а "Act" — впровадження змін, які випливають із висновків аудиту.

До початку інтеграції ШІ важливо пересвідчитись, що його застосування не порушує фундаментальних принципів аудиту ІБ, що включають чесність, об'єктивність, професіоналізм, конфіденційність та незалежність. ШІ, за умови належного навчання, може забезпечити високий рівень об'єктивності й незмінності у формуванні звітності. Водночас відповідальність за збереження конфіденційності вхідних і вихідних даних залишається за аудиторською групою. Незалежність алгоритмів ШІ від конфлікту інтересів забезпечує їх придатність для впровадження в аудиторські процеси.

Етап планування. Під час постановки цілей аудиту аудиторська група спирається на вимоги організації, стандарти ІБ (зокрема ISO/IEC 27001:2022), а також виявлені раніше ризики. ШІ може бути задіяний на цьому етапі для попереднього аналізу текстів політик, положень та інструкцій за допомогою технологій NLP. Це значно прискорює роботу з великими обсягами документації. За наявності постійного моніторингу ризиків ІБ у організації, можлива автоматизація збору таких даних, що дозволяє формувати контекст для ШІ. Водночас тут виникає ризик: якщо вхідні дані є хибними або неповними, результати аналізу можуть бути недостовірними. Цей аспект особливо критичний для навчання моделей, що використовуються в подальшому.

На *етапі оцінки ризиків* ШІ може агрегувати інформацію з різнорідних джерел, аналізуючи весь масив даних, а не лише його частину, як у класичних підходах. Такі можливості дозволяють виявляти закономірності та приховані залежності, що особливо важливо при оцінці комплексної моделі загроз. Водночас слід враховувати, що організація не завжди готова надавати повні набори даних через побоювання витоку або порушення конфіденційності. Формування програми аудиту це наступний крок, де ШІ може сформувати аналітичний звіт із розподілом ролей у команді, оцінкою навантаження, необхідними ресурсами тощо. Завдяки автоматизованому плануванню зменшується ймовірність помилок, спричинених людським чинником. Ініціалізація аудиту передбачає встановлення контакту з організацією та перевірку можливості його проведення. Хоч ШІ малоефективний у комунікаційних завданнях, він здатен оцінювати готовність організації до аудиту, спираючись на попередньо зібрані дані щодо доступності інформації, інфраструктури та персоналу. Під час підготовки до аудиту системи ШІ можуть проводити попередній аналіз об'ємних документів, класифікуючи їх за пріоритетністю, типами загроз або політиками контролю доступу. Також можливе використання ШІ для оптимізації розподілу обов'язків між членами аудиторської групи з урахуванням їхніх кваліфікацій та досвіду.

Етап виконання. На цьому етапі ШІ виступає активним аналітичним інструментом. Він може автоматично обробляти інформацію з внутрішніх систем організації: документи ІБ, конфігурації систем, мережеві журнали, плани приміщень, перелік активів та дані про користувачів. Частина інформації може бути зібрана автоматично через сканери вразливостей, SIEM-системи або засоби тестування на проникнення. Відповідно до мети аудиту, ШІ може перевіряти відповідність вимогам безпеки, виявляти потенційні канали витоку інформації, оцінювати ефективність технічних та організаційних засобів захисту, надавати рекомендації щодо покращення архітектури ІБ. генерувати концепції для побудови безпечного середовища.

Якість результатів значною мірою залежить від адекватності навчання моделей. Неправильно адаптовані алгоритми можуть спричинити помилки, тому постійна валідація ШІ-рішень має бути обов'язковою. Формування звіту за результатами аудиту також може бути автоматизованим. Якщо всі етапи були реалізовані із залученням ШІ, можливе створення структурованого звіту в режимі реального часу. В іншому разі ШІ може зібрати дані, згенеровані аудиторською групою, і перетворити їх у стандартизований формат. В обох випадках фінальний документ повинен бути перевірений аудиторською командою.

Наприклад, у корпоративній інформаційній системі підприємства, що використовує централізовану систему моніторингу подій безпеки, ШІ може бути інтегрований для аналізу поведінки користувачів та мережевого трафіку. У процесі нормальної роботи система формує профіль типової активності кожного користувача, враховуючи часові характеристики входу в систему, частоту звернення до ресурсів, обсяг переданих даних та характер взаємодії з інформаційними активами. У разі виникнення відхилень від сформованого профілю, наприклад, коли обліковий запис співробітника, який зазвичай працює у робочий час, починає здійснювати активні підключення до внутрішніх серверів у нічний період, система фіксує аномалію. Додатковим індикатором ризику може бути різке збільшення обсягу вихідного трафіку або спроби доступу до ресурсів, які не входять до звичайної зони відповідальності користувача. На основі сукупності таких факторів алгоритми машинного навчання класифікують поведінку як потенційно небезпечну. У подібній ситуації інтелектуальна система може автоматично ініціювати формування інциденту безпеки, передати відповідне повідомлення адміністратору або навіть тимчасово обмежити доступ до критичних ресурсів до моменту підтвердження легітимності дій користувача. Важливо, що аналіз здійснюється не за жорстко заданими правилами, а з урахуванням поведінкових характеристик, що дозволяє виявляти раніше невідомі сценарії атак, зокрема внутрішні загрози або компрометацію облікових записів. У традиційному аудиті ІБ подібні інциденти, як правило, можуть бути виявлені лише постфактум під час детального аналізу журналів подій, що потребує значних часових витрат і залучення експертів. Використання ж ШІ дозволяє здійснювати безперервний моніторинг та оперативно реагувати на відхилення у режимі реального часу, що суттєво підвищує ефективність виявлення загроз і знижує ризик витоку інформації. Таким чином, застосування інтелектуальних алгоритмів у процесі аудиту дає можливість перейти від реактивного аналізу інцидентів до проактивного виявлення загроз на основі поведінкових моделей, що є особливо важливим у сучасних умовах зростання складності кіберзагроз.

Етап перевірки та вдосконалення. Після завершення аудиту необхідно оцінити ефективність проведеної роботи. ШІ може аналізувати виконання цілей, відповідність плану, якість зібраних доказів та зворотний зв'язок від зацікавлених сторін. У разі виявлення відхилень ШІ може генерувати пропозиції щодо вдосконалення програми аудиту. Під час внутрішнього аудиту або регулярного моніторингу систем ІБ, системи ШІ можуть виконувати постійний контроль за індикаторами безпеки та в режимі реального часу оновлювати модель ризиків організації. Таким чином, цикл PDCA може бути реалізований у безперервному автоматизованому режимі.

На основі проведеного дослідження можна зробити висновок, що застосування ШІ на різних етапах аудиту ІБ є не лише доцільним, а й потенційно надзвичайно ефективним. Інтелектуальні системи можуть виступати як засіб автоматизації рутинних операцій, так і інструмент експертної підтримки в ухваленні рішень, забезпечуючи об'єктивний і неупереджений аналіз, що особливо важливо в умовах зростання складності та обсягу цифрових активів. Регулярне проведення як зовнішнього, так і внутрішнього аудиту з використанням ШІ забезпечує стабільне функціонування критичних бізнес-процесів і підвищує рівень цифрової стійкості організації. Для досягнення такого рівня ефективності організаціям доцільно фокусуватися на поступовій інтеграції ШІ у свою інформаційну інфраструктуру, впровадженні сумісних рішень та адаптації кадрового складу до роботи з новими технологіями. Успішна імплементація таких систем вимагає цілеспрямованої підготовки працівників як у внутрішніх службах безпеки, так і в аудиторських компаніях з акцентом на цифрові компетентності, управління ризиками ШІ та етичні аспекти його використання. У перспективі використання інтелектуальних систем моніторингу на базі ШІ може частково або повністю замінити необхідність у періодичних внутрішніх аудитах, здійснюючи автоматизований аналіз подій безпеки в режимі реального часу. Це дозволить аудиторським службам перейти від реактивної моделі контролю до проактивного підходу з фокусом на запобігання загрозам. У підсумку, інтеграція ШІ в аудит ІБ має всі передумови для того, щоб стати стандартом сучасної аудиторської практики.

УДК 004.8:004.056

П. В. Куріщенко *ст.2 курсу*, Л. В. Константинова
kurishchenkopavo@kntu.kr.ua, liliyashel1976@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький, Україна

ЗАСТОСУВАННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ БАЗ ДАНИХ

У сучасних інформаційних системах бази даних (БД) є критично важливим компонентом, що забезпечує зберігання та обробку великих обсягів персональних і корпоративних даних [1]. Через зростання цифровізації та ускладнення кіберзагроз традиційні методи захисту - такі як контроль доступу, автентифікація та журналювання - часто не здатні протистояти атакам "нульового дня", SQL-ін'єкціям та внутрішнім загрозам, що маскуються під легітимні операції [2,3] і тому виникає завдання для проведення досліджень з застосуванням методів з впровадженням штучного інтелекту (ШІ) для безпечної роботи БД.

Бази даних залишаються привабливою мішенню для зловмисників через високу концентрацію конфіденційної інформації. До найбільш критичних загроз належать:

– SQL-ін'єкції (SQLi): впровадження шкідливого коду у запити для несанкціонованого доступу або зміни даних [3];

– Внутрішні загрози: зловживання правами доступу адміністраторами або привілейованими користувачами [2,4];

– Компрометація облікових записів: використання викрадених даних для несанкціонованого доступу або витоку інформації [3,4].

Класичні механізми захисту баз даних, які орієнтуються на статичні правила та сигнатури, не дозволяють ефективно адаптуватися до нових типів атак або виявляти складні аномалії в поведінці користувачів [2]. Перспективним рішенням є впровадження методів машинного навчання (ML) та інтелектуального аналізу даних (Data Mining). Методи кластеризації та нейронних мереж можуть автоматизувати обробку логів (тобто файлів з спеціальною інформацією системи) БД, виявляючи нетипові шаблони поведінки та потенційні загрози [1,3]. Кластеризаційні алгоритми (наприклад, k-means або DBSCAN) дозволяють групувати схожі дії користувачів і виділяти аномалії, тоді як нейронні мережі (включно з автоенкодерами та глибокими моделями) здатні виявляти складні, нелінійні залежності у великих наборах логів.

Основні напрями застосування ШІ у захисті БД включають:

1. **Поведінковий аналіз (UEBA):** формування профілів нормальної активності користувачів і виявлення відхилень, наприклад нетипових обсягів вивантаження даних або доступів у неробочий час [4].

2. **Інтелектуальний аналіз SQL-запитів:** ML-методи дозволяють визначати підозрілі запити навіть без наявності відомих сигнатур SQL-ін'єкцій [3].

3. **Виявлення аномалій:** системи на основі аномалій перевершують сигнатурні підходи у виявленні раніше невідомих загроз, оскільки вони вивчають поведінку системи, а не тільки базу відомих атак [2].

Для досягнення максимального рівня захисту рекомендується поєднувати інтелектуальний аналіз із надійними криптографічними протоколами. Сучасна криптографія забезпечує фундамент конфіденційності, тоді як методи ШІ додають адаптивний рівень, який дозволяє визначати аномальні дії та потенційну компрометацію ключів [5].

Висновки. Аналіз підтверджує, що інтеграція методів штучного інтелекту є необхідною умовою для еволюції систем захисту баз даних. Використання ML-моделей дозволяє: автоматизувати виявлення інцидентів і зменшити навантаження на адміністраторів; знизити вплив людського фактора завдяки об'єктивному аналізу великих даних; забезпечити адаптивність до нових типів атак завдяки здатності алгоритмів до самонавчання.

Таким чином, перехід від статичних моделей захисту до інтелектуальних систем є важливим кроком для забезпечення цілісності, доступності та конфіденційності інформації в умовах сучасних кіберзагроз.

Список літератури

1. ResearchGate, Survey of Machine Learning Methods for Database Security, 2020.
2. P. Garcia-Teodoro, J. Diaz-Verdejo, et al., Anomaly-based network intrusion detection: Techniques, systems and challenges, 2009.
3. MDPI, Deep Learning Approaches for SQL Injection Detection, 2025.
4. W. Xu, W. Jiang, et al., UEBA for Enterprise Security: Behavior Analytics for Threat Detection, 2021.
5. W. Stallings, Cryptography and Network Security: Principles and Practice, Pearson Education, 2017.

УДК 004.056, 004.75

О.М. Дрєєв, Р. Лук'яненко
drey.sanya@gmail.com, klank-07@ukr.net
Центральноукраїнський національний технічних університет, Кропивницький

ПОРІВНЯННЯ АЛГОРИТМІВ ШИФРУВАННЯ МЕРЕЖНОГО ТРАФІКУ В СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Темпи розвитку інформаційних технологій росте з кожним роком. Відповідно, розширюється структура мережі інтернет речей, що вливає на збільшення обсягів трафіку. Також пристрої стають «розумнішими», що призводить до більш складних та об'ємних повідомлень в мережі інтернет речей. Але й на тлі зростання обсягів трафіку спостерігається значне збільшення вразливих пристроїв. Наприклад [15], iRobot (роботи-пилососи Roomba J7) збирали приватні фотографії для навчання штучної нейронної мережі. Цей випадок демонструє, що навіть за відсутності прямого злому системи витік даних може відбуватися через неналежну організацію обробки інформації. З огляду на ріст трафіку інтернет речей та збільшення вимог до захисту приватних даних, виникає потреба постійного покращення методів захисту даних в цих системах, що стосується не тільки потужності таких методів а й їх надійності та ефективності. У такому контексті питання вибору алгоритму шифрування мережного трафіку для інформаційної системи стоїть особливо гостро. В свою чергу, розглядаючи варіанти відповідні особливим умовам, наприклад висока потреба у швидкодії чи обмежені можливості розрахункової потужності, дане питання практично виявляється критичним. Таким чином система на мікроконтролерному керуванні з можливостями зв'язку потребує надійності, що встає в суперечність до наявних ресурсів, як обчислювальних, так і вимог до пам'яті.

Метою дослідження, як вже було визначено вище, є порівняння сучасних алгоритмів шифрування за критеріями надійності, швидкодії та ресурсомісткості. Часто використання великих та громістких криптографічних алгоритмів потребують виявляються занадто ресурсоємкими для певних систем, в той час як інші, занадто повільними. Нижче приведена таблиця, в якій визначено десять популярних алгоритмів шифрування мережного трафіку і оцінка деяких їх характеристик, таких як надійності, обчислювальної складності а також вимог до пам'яті. Наприклад, мікроконтролери серії ESP32 мають обсяг пам'яті 520Kb, але при використанні мережевих протоколів, доступний осяг пам'яті зменшується до 100-200Кб [13], що підтверджено на практиці. Використання менш потужних мікроконтролерів ускладнює і оцінки використання пам'яті та дає ще менші показники доступної пам'яті мікроконтролера.

Результати випробувань класичних реалізацій алгоритмів не є унікальними і доступні для загального користування, наприклад [14]. Основні тлумачення бенчмарків наведено в таблиці:

Таблиця
Порівнянна алгоритмів шифрування мережного трафіку

№	Алгоритм	Тип	Надійність	Обчислювальна складність	Пам'ять
1	AES-256	Симетричний	Дуже висока	Середня	Низькі
2	AES-128	Симетричний	Висока	Низька	Низькі
3	ChaCha20	Симетричний (потоковий)	Дуже висока	Низька	Дуже низькі
4	Twofish	Симетричний	Дуже висока	Середня	Середні
5	Camellia	Симетричний	Дуже висока	Середня	Середні
6	Blowfish	Симетричний	Середня	Середня	Низькі
7	RSA-4096	Асиметричний	Висока	Дуже висока	Високі
8	RSA-2048	Асиметричний	Висока	Висока	Середні
9	ECC (Curve25519)	Асиметричний	Дуже висока	Низька	Дуже низькі
10	3DES	Симетричний	Низька (застарілий)	Висока	Низькі

Експериментальні дослідження показують, що на мікроконтролерах класу ESP32 швидкість симетричних алгоритмів суттєво залежить від наявності апаратного прискорення. Зокрема, AES-256 демонструє продуктивність близько 1.5MB/s у програмній реалізації та до 10MB/s при використанні апаратного криптографічного модуля, тоді як ChaCha20 досягає приблизно 3MB/s без апаратної підтримки. Це обумовлює перевагу ChaCha20 у вбудованих системах без криптоприскорювачів. Тому для обрання конкретного алгоритму потрібно враховувати не лише криптостійкість алгоритму, але й наявність апаратних прискорювачів обчислення у обраній апаратній платформі. Тому, з приведеним вище аналізом, а також з описаними вище твердженнями, вибір алгоритму суттєво залежить від технічних обмежень системи. Найбільш ресурсномістким

є RSA-4096 має найвищу обчислювальну складність та високі вимоги до пам'яті, що варто використовувати лише в критичних до криптостійкості системах для обміну ключами та аутентифікації. А також маємо ряд застарілих рішень, таких як алгоритм 3DES, що має низьку надійність та високу складність обчислень, що робить його найменш придатним у використанні в сучасних системах.

У сучасних протоколах, зокрема Transport Layer Security, застосовується гібридний підхід. Алгоритми асиметричної криптографії, такі як Elliptic Curve Cryptography, використовуються для обміну ключами. Після чого симетричні алгоритми забезпечують ефективне шифрування основного обсягу даних. Це забезпечує підвищену надійність обміну ключами, а також заощаджуються ресурси на шифрування та дешифрування трафіку.

Висновок. Сучасні протоколи захисту мережевого трафіку, зокрема TLS 1.3, використовують гібридний підхід: асиметричні алгоритми (наприклад, ECC) застосовуються для обміну ключами, тоді як симетричні (AES або ChaCha20) – для основного шифрування даних. Це дозволяє досягти оптимального балансу між безпекою та продуктивністю. Тобто:

1. Симетричні алгоритми є основним засобом шифрування мережевого трафіку.
2. Асиметричні алгоритми використовуються для безпечного обміну ключами.
3. Алгоритм ChaCha20 є ефективнішим у програмній реалізації, тоді як AES – при наявності апаратного прискорення.
4. Застарілі алгоритми, такі як Triple DES, не рекомендуються до використання.
5. Оптимальний вибір алгоритму залежить від обчислювальних ресурсів системи.

Для мікроконтролерних систем інтернету речей, зокрема ESP32-WROOM-32 DevKit V1, доцільно використовувати алгоритми шифрування, які враховують наявність апаратного прискорення. Зокрема, алгоритм AES є оптимальним вибором при використанні вбудованого криптографічного модуля, що забезпечує високу продуктивність і низьке енергоспоживання. Проте, для реалізації власних протоколів є доцільним використання ChaCha20, який демонструє високу ефективність у програмній реалізації.

Список літератури

1. NIST. FIPS PUB 197: Advanced Encryption Standard (AES). – 2001.
2. NIST SP 800-57. Recommendation for Key Management.
3. NIST SP 800-131A. Transitioning the Use of Cryptographic Algorithms and Key Lengths.
4. Schneier B. Applied Cryptography. – Wiley, 1996.
5. Paar C., Pelzl J. Understanding Cryptography. – Springer, 2010.
6. Bernstein D. ChaCha, a variant of Salsa20.
7. Bernstein D., Lange T. SafeCurves: choosing secure curves for elliptic-curve cryptography.
8. Lenstra A., Verheul E. Selecting Cryptographic Key Sizes.
9. RFC 8439. ChaCha20 and Poly1305 for IETF Protocols.
10. RFC 7748. Elliptic Curves for Security.
11. OpenSSL Project. Performance benchmarks of cryptographic algorithms.
12. Langley A. TLS performance analysis and ChaCha20 adoption.
13. Minimizing RAM Usage. URL: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/performance/ram-usage.html
14. Crypto Benchmark on ESP32 MCU Tensilica Xtensa LX6 @ 240 MHz URL: https://www.oryx-embedded.com/benchmark/espressif/crypto-esp32.html
15. Roomba Test Robot Captures Private Images, Leading to Global Privacy Breach. URL: https://oecd.ai/en/incidents/2022-12-19-bd31

УДК 004.056.5

К.В. Навроцька[1], Н.С. Петляк[1]
katia43060@gmail.com, npetlyak@khmnu.edu.ua
[1]Хмельницький національний університет, Хмельницький

МЕТОД ВИЯВЛЕННЯ ОБЛІКОВИХ ЗАПИСІВ З АНОМАЛЬНОЮ АКТИВНІСТЮ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ

У сучасних інформаційно-комунікаційних системах проблема забезпечення безпеки облікових записів набуває особливої актуальності у зв'язку зі зростанням кількості кібератак, спрямованих на компрометацію автентифікаційних даних користувачів. Використання легітимних облікових записів зловмисниками є одним із найскладніших для виявлення сценаріїв, оскільки така активність часто не порушує формальних правил доступу. Особливої складності набуває виявлення подібних інцидентів у середовищах із динамічною інфраструктурою, зокрема в програмно-керованих мережах, де централізоване управління та висока гнучкість створюють додаткові можливості для прихованої діяльності. Традиційні підходи, засновані на сигнатурному аналізі або статичних правилах, не забезпечують належного рівня ефективності у виявленні нових або модифікованих атак. Це зумовлює необхідність розроблення нових методів аналізу поведінки користувачів, які б враховували контекст і характер їхніх дій навіть за відсутності явних порушень політик безпеки. Інтеграція поведінкових та часових характеристик у межах єдиної моделі дозволяє створити адаптивний механізм виявлення аномалій, що суттєво підвищує достовірність ідентифікації прихованих загроз.

Метод виявлення облікових записів з аномальною активністю доцільно будувати на основі інтеграції поведінкових характеристик користувачів та часових метрик їхньої діяльності, що дозволяє підвищити точність виявлення інцидентів інформаційної безпеки та зменшити рівень хибнопозитивних спрацювань. Основою такого підходу є формування індивідуального профілю користувача, який відображає типові шаблони його взаємодії з інформаційною системою, а також часові закономірності виконання операцій.

У межах реалізації методу здійснюється збір даних із журналів автентифікації, мережевих сесій, історії виконання команд та інформації про ролі користувачів. На основі цих даних формується вектор ознак користувача x_i, який описує його поведінкову активність $x_i = (x_1, x_2, \dots, x_n)$, де кожна компонента відповідає певній характеристиці, наприклад інтенсивності сесій, кількості унікальних IP-адрес або частоті виконання команд.

Такий вектор дозволяє формалізувати поведінку користувача у вигляді багатовимірного простору ознак, що є необхідним для подальшого кількісного аналізу.

Для визначення нормальної поведінки використовується статистичний профіль, який задається вектором середніх значень μ та стандартних відхилень σ, отриманих на основі історичних даних. Відхилення поточного значення ознаки від її типової величини оцінюється за допомогою z-оцінки:

$$z_j = \frac{x_j - \mu_j}{\sigma_j}$$

Ця величина дозволяє нормалізувати різнорідні ознаки до єдиної шкали, що є важливим у випадку, коли характеристики мають різну природу та одиниці вимірювання. Крім того, використання z-оцінки дає змогу інтерпретувати відхилення у термінах статистичної значущості, тобто визначити, наскільки поточне значення є нетиповим відносно історичної поведінки користувача.

Інтегральна оцінка поведінкової аномальності визначається як агрегована функція відхилень:

$$S_b = \frac{1}{n}\sum_{j=1}^{n} |z_j|$$

Необхідність введення такої оцінки зумовлена тим, що окремі ознаки самі по собі можуть не свідчити про аномалію, однак їх сукупне відхилення формує характерний шаблон підозрілої поведінки. Агрегація дозволяє отримати узагальнену числову характеристику, яка відображає загальний ступінь відхилення поведінки користувача від його нормального профілю. Використання середнього абсолютного значення забезпечує однаковий внесок кожної ознаки та запобігає взаємній компенсації позитивних і негативних відхилень.

Паралельно здійснюється аналіз часових характеристик активності користувача. На основі історичних даних будується ймовірнісний розподіл активності протягом доби, який описується функцією $P(h)$, де h - година доби. Оцінка часової аномальності визначається як $S_t = 1 - P(h_t)$, де h_t - фактичний час виконання дії. Запровадження цієї метрики дозволяє кількісно оцінити, наскільки поточна активність відповідає типовому часовому шаблону користувача. Якщо користувач зазвичай працює у денний час, то нічна активність матиме низьку ймовірність і, відповідно, високу оцінку аномальності. Таким чином, дана складова враховує контекст виконання дій, що є критично важливим для виявлення компрометації облікових записів.

Для прийняття узгодженого рішення необхідно інтегрувати поведінкову та часову складові в єдиний

показник. Це реалізується за допомогою зваженої лінійної комбінації $S = \alpha * S_b + \beta * S_t$.

Введення вагових коефіцієнтів α та β дозволяє регулювати вплив кожного типу ознак на кінцевий результат. Це є важливим, оскільки в різних сценаріях атаки домінуючу роль можуть відігравати різні фактори: у деяких випадках аномальною є саме поведінка, тоді як у інших - час виконання дій. Таким чином, зважена агрегація забезпечує гнучкість моделі та можливість її адаптації до специфіки середовища.

Остаточне рішення щодо наявності аномальної активності приймається шляхом порівняння інтегральної оцінки з пороговим значенням:

$$Alert = \begin{cases} 1, \text{якщо } S > 0 \\ 0, \text{інакше.} \end{cases}$$

Запропонований метод демонструє підвищену достовірність у випадках, коли поведінкова активність користувача суттєво відхиляється від сформованого профілю та одночасно фіксується порушення часових закономірностей виконання операцій. Зокрема, у ситуації, коли обліковий запис адміністратора використовується для виконання нетипових дій, таких як запуск незвичних команд, доступ до раніше нехарактерних ресурсів або зміна параметрів системи, у поєднанні з активністю у часові інтервали, що не відповідають типовому режиму роботи даного користувача, інтегральна оцінка аномальності перевищує встановлений пороговий рівень. Це, у свою чергу, ініціює механізм реагування та сигналізує про потенційний інцидент інформаційної безпеки. Застосування зваженої агрегації дозволяє адаптувати модель до специфіки конкретного середовища, враховуючи, що в різних сценаріях атак домінуючу роль можуть відігравати або поведінкові фактори, або часовий контекст. Такий підхід мінімізує вплив випадкових відхилень, які не є ознаками цілеспрямованого зламу, та забезпечує стабільність роботи системи моніторингу.

Практична реалізація запропонованого методу передбачає його інтеграцію до складу систем моніторингу інформаційної безпеки, зокрема SIEM-систем, систем управління доступом або платформ аналізу мережевого трафіку. Метод може бути реалізований як окремий аналітичний модуль, що функціонує у режимі реального часу або періодичної обробки подій, отримуючи дані з різних джерел інформаційно-комунікаційної системи. Вхідними даними для функціонування методу є журнали автентифікації, записи про мережеві з'єднання, історія виконання команд, інформація про використані ресурси, IP-адреси, часові мітки подій, а також атрибути ролей і прав доступу користувачів. Дані можуть надходити як із внутрішніх систем (сервери, контролери домену, додатки), так і з мережевих пристроїв (маршрутизатори, міжмережеві екрани). Перед обробкою вони проходять етап нормалізації та агрегації для формування уніфікованого вектору ознак.

Результатом роботи є інтегральна оцінка аномальності для кожного облікового запису або окремої сесії користувача. У разі перевищення встановленого порогового значення формується подія безпеки, яка передається до системи реагування та безпосередньо адміністраторам безпеки. Додатково можуть формуватися пояснювальні характеристики (наприклад, які саме ознаки дали найбільший внесок в аномалію), що підвищує інтерпретованість результатів. Отримані результати надходять до центрів моніторингу безпеки або відповідальних адміністраторів, де здійснюється подальший аналіз інциденту. Реакція на виявлену аномалію може бути як автоматизованою (блокування облікового запису, завершення сесії, вимога повторної автентифікації), так і напівавтоматичною з участю аналітика. Вибір сценарію реагування залежить від критичності ресурсу, рівня аномалії та політик безпеки організації. Важливою перевагою такого підходу є здатність виявляти складні та приховані сценарії компрометації облікових записів, зокрема у випадках використання легітимних облікових даних зловмисниками. На відміну від традиційних методів контролю доступу, які орієнтовані на перевірку автентичності користувача, запропонована модель аналізує контекст та характер його дій, що дозволяє виявляти відхилення навіть за відсутності явних порушень політик безпеки. Це є особливо актуальним для виявлення прихованих каналів витоку інформації, які можуть реалізовуватися через легітимні інтерфейси доступу та не супроводжуватися типовими ознаками атак. Таким чином, інтеграція поведінкових та часових характеристик у межах єдиної аналітичної моделі забезпечує формування більш точного та адаптивного механізму виявлення аномальної активності. Такий підхід дозволяє одночасно враховувати як зміст виконуваних дій, так і контекст їх реалізації у часі, що істотно підвищує достовірність ідентифікації загроз. Це є важливим для сучасних інформаційно-комунікаційних систем, зокрема програмно-керованих мереж, де висока динамічність середовища та централізований характер управління створюють додаткові виклики для забезпечення інформаційної безпеки. Використання статистичних профілів та z-оцінок дозволяє ефективно нормалізувати різнорідні дані, приводячи їх до єдиної шкали для кількісного аналізу.

УДК 004.056.5:621.39

О.М. Дрєєв, Д. Паращенко
drey.sanya@gmail.com, denus.parashencko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ КОРПОРАТИВНОЇ ЗАХИЩЕНОЇ IOT МЕРЕЖІ З ВИКОРИСТАННЯМ «ZERO TRUST» ТЕХНОЛОГІЙ

Зі стрімким розвитком технологій Інтернету речей питання забезпечення безпеки корпоративних IoT-мереж набуває особливої актуальності. Зростаюча кількість підключених пристроїв та їхні обмежені обчислювальні ресурси створюють ризики несанкціонованого доступу, кіберінцидентів та витоку даних. Особливо важливим є забезпечення захисту IoT-мереж, у яких функціонують пристрої, критичні для безпеки обладнання та здоров'я людини. Водночас необхідно забезпечувати й масштабованість системи IoT-пристроїв, оскільки додавання та вилучення окремих пристроїв відбувається порівняно часто. В умовах частої зміни парку пристроїв актуальними стають загрози підміни даних у мережі, реєстрації підроблених пристроїв та перехоплення конфіденційної інформації. Тому метою дослідної роботи є вибір технологій захисту корпоративної мережі від несанкціонованого втручання в роботу системи.

За основний принцип побудови системи взаємодії IoT-пристроїв пропонується використовувати «Zero Trust» технології, що ґрунтуються на принципі «Не довіряй нікому, перевіряй усіх». У межах цього підходу кожен пристрій і користувач розглядається як потенційна загроза, а доступ надається лише після проходження автентифікації та авторизації для кожного запиту. Актуальність «Zero Trust» технології підтверджується тим, що підроблені пристрої не мають можливості ініціювати сесію зв'язку чи отримати доступ до ресурсів мережі, навіть якщо для некритичних даних використовується відкрита передача задля заощадження обчислювальних ресурсів мікроконтролера.

Основними відмінностями корпоративної IoT мережі від систем розумного будинку є масштабованість та вимоги до безпеки. Ці відмінності випливають з призначення мереж: домашні мережі призначені для забезпечення зручності експлуатації, легкості підключення додаткових пристроїв; для корпоративної мережі на перший план виходять вимоги надійності та захищеності зв'язку між пристроями при значно більших масштабах. Більш детальне порівняння наведено в наступній таблиці:

Таблиця
Порівняння властивостей домашніх та корпоративних IoT мереж

№	Назва	Розумний дім	Корпоративна мережа
1	Масштаб та кількість пристроїв	Розумний будинок зазвичай обслуговує до сотні пристроїв.	Тисячі або десятки тисяч пристроїв, які потребують високої пропускної спроможності мережі.
2	Надійність та відмовостійкість	Відмова пристрою не є критичною, окремі відповідальні одиниці часто мають можливість працювати незалежно в автономному режимі.	Вимагається надійна безперервна робота великої кількості пристроїв, де збій може нанести величезні збитки.
3	Безпека	Часто обмежується паролем WiFi	Корпоративний IoT має багаторівневий захист, захищені канали зв'язку та оновлення безпеки.
4	Протоколи та інтеграція	Wi-Fi, Bluetooth, Zigbee	NB-IoT, LoRaWAN, OPC UA
5	Обслуговування	В більшості користувач налаштовує зв'язок самостійно.	Корпоративні IoT системи потребують постійної професійної інженерної підтримки.

У даній роботі також досліджується роль логування подій. Логування використовується для машинного аналізу даних. Також пропонується проведення виявлення підозрілої активності в системі безпеки на основі алгоритмів кластеризації Isolation Forest, який може використовуватися для пошуку аномалій в даних. Використання «Zero Trust» технології разом з аналізом логів дозволяє не лише посилити достовірність джерел інформації, але й за зміною активності пристроїв в мережі виявляти атаки та втручання в роботу системи, що дозволить вчасно виявляти та усувати загрози. Крім того, розглядається принцип мінімізації привілеїв, що обмежує доступ користувачів і пристроїв лише до необхідних ресурсів, що підвищить захист від кіберінцидентів. Базовим механізмом є використання тимчасових токенів автентифікації замість статичних

паролів. Після входу користувач отримує токен доступу з обмеженим терміном дії, що буде унікальним для кожного запиту. Передача даних здійснюється через TLS-канал із застосуванням сесійних ключів.

Система централізовано контролює показники роботи мережі: у ній передбачено інвентаризацію дозволених пристроїв, контроль мережевих портів та аналіз трафіку. Доступ до мережі буде надаватися виключно зареєстрованим пристроям після підтвердження адміністратором.

Висновок. У роботі запропоновано архітектуру корпоративної IoT-мережі, побудовану на принципах «Zero Trust». Поєднання тимчасових токенів автентифікації, TLS-шифрування з сесійними ключами, принципу мінімізації привілеїв та централізованого контролю доступу забезпечує багаторівневий захист від несанкціонованого втручання. Застосування алгоритму кластеризації Isolation Forest для аналізу логів дозволяє виявляти аномальну активність пристроїв та оперативно реагувати на потенційні загрози. Запропонований підхід є масштабованим і може бути адаптований для корпоративних мереж різного рівня складності, що робить його перспективним для подальшого впровадження та дослідження.

Список літератури

1. He, Yuanhang, Huang, Daochao, Chen, Lei, Ni, Yi, Ma, Xiangjie, A Survey on Zero Trust Architecture: Challenges and Future Trends, Wireless Communications and Mobile Computing, 2022, 6476274, 13 pages, 2022. https://doi.org/10.1155/2022/6476274URL: https://onlinelibrary.wiley.com/doi/full/10.1155/2022/6476274

2. Innocent Uzougbo Onwuegbuzie, Alabi Oyegbola Augustine A Review of Authentication and Authorization Mechanisms in Zero Trust Architecture: Evolution and Efficiency, Tech-Sphere Journal for Pure and Applied Sciences Vol. 2 No. 1 (2025), Published 04-04-2025, https://doi.org/10.5281/zenodo.15149866 URL: https://stem.techspherejournals.com/index.php/tsjpas/article/view/1/1

3. Alshomrani, Shrooq, Li, Shancang, PUFDCA: A Zero-Trust-Based IoT Device Continuous Authentication Protocol, Wireless Communications and Mobile Computing, 2022, 6367579, 9 pages, 2022. https://doi.org/10.1155/2022/6367579 URL: https://onlinelibrary.wiley.com/doi/full/10.1155/2022/6367579

4. B. Bamleshwar Rao, Dr. Akhilesh A. Waoo Desing a novel approach for token based authentication in IOT networks Ilkogretim Online - Elementary Education Online, Year; Vol 20 (Issue 4): pp. 2401-2406, doi: 10.17051/ilkonline.2021.04.275 URL: https://www.researchgate.net/profile/Akhilesh-Waoo-2/publication/357157112_DESIGN_A_NOVEL_APPROACH_FOR_TOKEN_BASED_AUTHENTICATION_IN_IOT_NETWORKS/links/61bdf9234b318a6970eed1c0/DESIGN-A-NOVEL-APPROACH-FOR-TOKEN-BASED-AUTHENTICATION-IN-IOT-NETWORKS.pdf

5. Gunasekaran Manogaran, Bharat S. Rawal, Vijayalakshmi Saravanan, Priyan M K, Qin Xin, and P. Shakeel. 2023. Token-Based Authorization and Authentication for Secure Internet of Vehicles Communication. ACM Trans. Internet Technol. 22, 4, Article 90 (November 2022), 20 pages. https://doi.org/10.1145/3491202

6. P. Rujichaikul and I. Rassameeroj, "Token-Based Authentication Monitoring System," in *Journal of Cyber Security and Mobility*, vol. 14, no. 4, pp. 777-798, July 2025, doi: 10.13052/jcsm2245-1439.1441 URL: https://ieeexplore.ieee.org/abstract/document/11203932

7. Jingcheng Yang, Enze Wang, Jianjun Chen and other Token Time Bomb: Evaluating JWT Implementations for Vulnerability Discovery, Network and Distributed System Security (NDSS) Symposium 2026, 23-27 February 2026, San Diego, CA, USA, ISBN 979-8-9919276-8-0, https://dx.doi.org/10.14722/ndss.2026.240697. URL: https://eki.im/assets/paper/jwtbomb-paper.pdf

8. N. Sarwar, I. S. Bajwa, M. Z. Hussain, M. Ibrahim and K. Saleem, "IoT Network Anomaly Detection in Smart Homes Using Machine Learning," in *IEEE Access*, vol. 11, pp. 119462-119480, 2023, doi: 10.1109/ACCESS.2023.3325929. URL: https://ieeexplore.ieee.org/abstract/document/10287977

9. S. A. Abdulkareem, C. Heng Foh, M. Shojafar, F. Carrez and K. Moessner, "Network Intrusion Detection: An IoT and Non IoT-Related Survey," in IEEE Access, vol. 12, pp. 147167-147191, 2024, doi: 10.1109/ACCESS.2024.3473289.URL: https://ieeexplore.ieee.org/abstract/document/10704655

10 Sudhir Chitnis, Neha Deshpande, Arvind Shaligram An Investigative Study for Smart Home Security: Issues, Challenges and Countermeasures, Wireless Sensor Network, 2016, 8, 61-68, http://dx.doi.org/10.4236/wsn.2016.84006.

11 S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez and B. Rubinstein, "Machine Learning in Network Anomaly Detection: A Survey," in *IEEE Access*, vol. 9, pp. 152379-152396, 2021, doi: 10.1109/ACCESS.2021.3126834. URL: https://ieeexplore.ieee.org/abstract/document/9610045

12. A. B. Nassif, M. A. Talib, Q. Nasir and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," in *IEEE Access*, vol. 9, pp. 78658-78700, 2021, doi: 10.1109/ACCESS.2021.3083060. URL: https://ieeexplore.ieee.org/abstract/document/9439459

13. Gerhard Munz, ¨ Sa Li, Georg Carle Traffic Anomaly Detection Using K-Means Clustering, Computer Networks and Internet Wilhelm Schickard Institute for Computer Science University of Tuebingen, Germany. URL: https://www.net.in.tum.de/projects/dfg-lupus/files/muenz07k-means.pdf

14. Jiuxing Zhou,Wei Fu,Wei Hu,Zhihong Sun, Tao He, Zhihong Zhang Challenges and Advances in Analyzing TLS 1.3-Encrypted Traffic: A Comprehensive Survey. Electronics 2024, 13(20), 4000; https://doi.org/10.3390/electronics13204000. URL: https://www.mdpi.com/2079-9292/13/20/4000

УДК 004.056.53 (045)

К.І. Тараненко, *здобувач кафедри кібербезпеки*[1] ,
А.В. Ільєнко, *канд. техн. наук, доцент, завідувач кафедри Кібербезпеки*[1],
О.В. Дубчак, *ст. викл. кафедри Кібербезпеки*[1]
8481263@stud.kai.edu.ua , anna.ilienko@npp.kai.edu.ua, 3915922@npp.kai.edu.ua
[1]*Державний університет «Київський авіаційний інститут», Київ, Україна*

СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ ОБ'ЄКТІВ КРИТИЧНОЇ ІНФРАСТРУКТУРИ: КЛАСИФІКАЦІЯ І ПРИНЦИПИ

Системи виявлення вторгнень є фундаментальним інструментом забезпечення кібербезпеки і пройшли значну еволюцію від сигнатурних підходів до сучасних рішень, що інтегрують машинне навчання та аналіз поведінки [1; 2]. Водночас зберігається актуальність базової парадигми виявлення зловживань і аномалій, а також потреба у гібридних архітектурах, що поєднують різні джерела даних і методи аналізу [3]. Особливої складності набуває застосування IDS у промислових середовищах, де необхідно враховувати специфіку технологічних процесів, обмеження ресурсів і критичність інфраструктури. Конвергенція інформаційних та операційних технологій (IT/OT) в рамках концепції Industrial Internet of Things суттєво розширює поверхню атаки об'єктів критичної інфраструктури (ОКІ), водночас унеможливлюючи застосування традиційних IT-орієнтованих засобів захисту без відповідної адаптації [4].

Метою роботи є аналіз та систематизація підходів до побудови систем виявлення вторгнень для об'єктів критичної інфраструктури, зокрема їх архітектурних моделей, методів виявлення та особливостей застосування в промислових середовищах, а також дослідження можливостей використання методів машинного навчання для підвищення ефективності IDS.

Системи виявлення вторгнень (Intrusion Detection Systems, IDS) являють собою програмно-апаратні комплекси, що здійснюють моніторинг мережевого трафіку, системних журналів або поведінки кінцевих точок з метою виявлення ознак несанкціонованого доступу чи порушення встановленої політики безпеки. Перший формальний опис IDS на основі статистичного профілювання нормальної поведінки запропонувала Denning [1], заклавши підґрунтя для подальшої еволюції галузі. Розгорнуту таксономію IDS, що охоплює виміри методу виявлення, архітектури розгортання, типу реакції та джерела даних, розробив Axelsson [2], і ця класифікація залишається стандартною системою координат у сучасних дослідженнях.

За розміщенням у мережевій архітектурі IDS поділяються на мережеві (NIDS) та хостові (HIDS). Мережеві IDS аналізують трафік у виділених точках мережевої інфраструктури через пасивне підключення (SPAN-порт або TAP-відгалужувач), забезпечуючи охоплення всього сегменту без агентів на кінцевих точках. Хостові IDS встановлюються безпосередньо на захищувані вузли й аналізують системні виклики, журнали подій і цілісність файлів. Для OT-середовищ ОКІ перевага надається пасивним NIDS, оскільки будь-які inline-рішення, що вносять затримку в мережевий трафік, є неприйнятними для промислових систем реального часу [3; 4].

Методи виявлення

Сигнатурний (знання-базований) метод базується на порівнянні мережевого трафіку або системних подій із базою відомих шаблонів атак (сигнатур, правил, індикаторів компрометації). Ключовою перевагою методу є відсутність хибнопозитивних спрацювань для відомих атак та висока швидкість класифікації. Критичним недоліком є принципова неспроможність виявити нові чи поліморфні атаки та атаки типу Living-off-the-Land, що використовують легітимні системні інструменти [3].

Аномалійний метод ґрунтується на формуванні профілю нормальної поведінки та виявленні статистично значущих відхилень від нього. Chandola, Banerjee та Kumar [5] систематизували підходи до виявлення аномалій за трьома типами: точкові, контекстуальні та колективні аномалії. Для OT-середовищ особливе значення мають контекстуальні аномалії, коли команда управління є технічно допустимою, але не відповідає поточному стану технологічного процесу. Sommer та Paxson [6] виокремили ключові виклики застосування методу в реальних умовах: «відкритий світ» нормального трафіку, дефіцит репрезентативних навчальних даних та неприйнятно високий рівень хибнопозитивних результатів.

Гібридний підхід поєднує обидва методи: сигнатурний шар забезпечує точну ідентифікацію відомих атак, тоді як аномалійний обробляє залишковий трафік для виявлення нових загроз. Порівняльний аналіз методів наведено у таблиці 1.

Таблиця 1

Порівняльний аналіз методів виявлення вторгнень для ОКІ

Критерій	Сигнатурний	Аномалійний	Гібридний
Виявлення відомих загроз	Відмінне	Задовільне	Відмінне
Виявлення нових загроз (0-day)	Відсутнє	Хороше	Хороше
Рівень хибнопозитивних	Низький	Високий	Середній
Вимоги до ресурсів	Низькі	Високі	Середні / Високі
Час навчання системи	Не потрібен	Тижні – місяці	Тижні – місяці
Пояснюваність рішень	Висока	Низька	Залежить від реалізації
Придатність для ОТ-мереж	Обмежена	Рекомендована	Оптимальна

IDS для промислових середовищ

Розгортання IDS у промислових мережах вимагає принципово іншого підходу, ніж у корпоративних IT-мережах. Hadžiosmanović зі співавторами [7] запропонували концепцію семантичного моніторингу безпеки, що виходить за межі аналізу мережевого трафіку та включає кореляцію мережевих команд із реальними значеннями технологічних параметрів. Авторами показано, що атаки, непомітні з точки зору мережевого трафіку, виявляються через відхилення фізичних параметрів від значень, визначених моделлю процесу.

Kleinmann та Wool [8] розробили метод точного моделювання поведінки Siemens S7 PLC на рівні SCADA-протоколу з використанням детермінованих скінченних автоматів. Підхід забезпечує практично нульовий рівень хибнопозитивних результатів для відомих нормальних патернів і дозволяє виявляти атаки, що порушують очікувані послідовності команд, навіть без знання сигнатур конкретних загроз. Принциповою відмінністю ICS-IDS від IT-IDS є необхідність декодування та семантичного аналізу промислових протоколів (Modbus, DNP3, IEC 61850, PROFINET), де аномалія може полягати не у нестандартній структурі пакета, а у технологічно неприпустимому значенні параметра команди [4].

Застосування машинного навчання

Buczak та Guven [9] у систематичному огляді понад 100 публікацій встановили, що ансамблеві методи (Random Forest, Gradient Boosting) стабільно демонструють найкращі результати на стандартних наборах даних, тоді як нейромережеві підходи є перспективними для виявлення складних поведінкових патернів. Shone зі співавторами [10] запропонували архітектуру на основі несиметричних глибоких автоенкодерів (NDAE) для навчання без учителя і показали значне підвищення точності порівняно з традиційними підходами.

Kim зі співавторами [11] застосували архітектуру LSTM для класифікації мережевих вторгнень, виявивши її ефективність для виявлення розподілених у часі аномалій, характерних для APT-угруповань. Mirsky зі співавторами [12] розробили систему Kitsune на основі ансамблю автоенкодерів, що навчається без заздалегідь визначених ознак і здатна до онлайн-виявлення аномалій у реальному часі з відносно низькою обчислювальною складністю — що є критичною вимогою для ОТ-середовищ із обмеженим апаратним забезпеченням.

Разом з тим, застосування машинного навчання в реальних IDS пов'язане з рядом невирішених проблем. Sommer та Paxson [6] вказали на фундаментальний виклик «відкритого світу»: моделі, навчені на обмеженому наборі даних атак, не здатні надійно узагальнюватися на весь можливий простір аномалій. Tavallaee зі співавторами [13] виявили серйозні статистичні вади широко використовуваного набору KDD CUP 99 (близько 78% дублікатів), що призводять до надмірно оптимістичних результатів оцінки моделей. Moustafa та Slay [14] розробили альтернативний набір UNSW-NB15 з реальним сучасним трафіком, а Sharafaldin зі співавторами [15] — CICIDS-2017, що наразі є більш репрезентативними орієнтирами для оцінки IDS.

Висновки. У результаті проведеного аналізу встановлено, що сучасні IDS класифікуються за методом виявлення, архітектурою розгортання, типом реакції та джерелами даних, причому найбільш

ефективними є гібридні підходи, що поєднують сигнатурні та аномалійні методи. Визначено, що сигнатурні методи забезпечують високу точність для відомих загроз, тоді як аномалійні дозволяють виявляти нові атаки, але супроводжуються підвищеним рівнем хибнопозитивних результатів. Показано, що для промислових середовищ доцільним є використання спеціалізованих підходів, зокрема семантичного моніторингу та моделей поведінки протоколів. Встановлено, що застосування методів машинного навчання, зокрема ансамблевих алгоритмів і глибоких нейронних мереж, забезпечує підвищення ефективності виявлення вторгнень, однак супроводжується проблемами якості навчальних даних, узагальнення моделей та відсутності стандартизованих підходів до оцінювання.

Список літератури

1. Denning D. E. An Intrusion-Detection Model. IEEE Transactions on Software Engineering. 1987. Vol. SE-13, No. 2. P. 222–232.

2. Axelsson S. Intrusion Detection Systems: A Survey and Taxonomy. Technical Report 99-15. Chalmers University of Technology. 2000. 35 p.

3. García-Teodoro P., Díaz-Verdejo J., Maciá-Fernández G., Vázquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security. 2009. Vol. 28, No. 1–2. P. 18–28.

4. Nicholson A., Webber S., Dyer S., Patel T., Janicke H. SCADA Security in the Light of Cyber-Warfare. Computers & Security. 2012. Vol. 31, No. 4. P. 418–436.

5. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey. ACM Computing Surveys. 2009. Vol. 41, No. 3. Article 15.

6. Sommer R., Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. IEEE Symposium on Security and Privacy. 2010. P. 305–316.

7. Hadžiosmanović D., Sommer R., Zambon E., Hartel P. H. Through the Eye of the PLC: Semantic Security Monitoring for Industrial Processes. ACSAC 2014. P. 126–135.

8. Kleinmann A., Wool A. Accurate Modeling of the Siemens S7 SCADA Protocol for Intrusion Detection and Digital Forensics. Journal of Digital Forensics, Security and Law. 2014. Vol. 9, No. 2. P. 37–50.

9. Buczak A. L., Guven E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials. 2016. Vol. 18, No. 2. P. 1153–1176.

10. Shone N., Ngoc T. N., Phai V. D., Shi Q. A Deep Learning Approach to Network Intrusion Detection. IEEE Transactions on Emerging Topics in Computational Intelligence. 2018. Vol. 2, No. 1. P. 41–50.

11. Kim J., Kim J., Thu H. L. T., Kim H. Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. PlatCon. 2016. P. 1–5.

12. Mirsky Y., Doitshman T., Elovici Y., Shabtai A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. NDSS 2018. P. 1–15.

13. Tavallaee M., Bagheri E., Lu W., Ghorbani A. A. A Detailed Analysis of the KDD CUP 99 Data Set. IEEE Symposium on Computational Intelligence for Security and Defense Applications. 2009. P. 1–6.

14. Moustafa N., Slay J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems. MilCIS 2015. P. 1–6.

15. Sharafaldin I., Habibi Lashkari A., Ghorbani A. A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. ICISSP 2018. P. 108–116.

УДК 004.056, 004.8

І.А. Лисенко, В.А. Резніченко
min_max@i.ua„ upsbilly@meta.ua
Центральноукраїнський національний технічний університет, Кропивницький

СУЧАСНІ МЕТОДИ DATA MINING ДЛЯ АНАЛІЗУ КІБЕРЗАГРОЗ

Сучасний рівень розвитку інформаційних систем призводить і до зростання складності кіберзагроз. Це пов'язано із розвитком хмарних технологій, інтернету речей та кіберфізичних систем, що обумовлює необхідність використання інтелектуальних методів аналізу даних (Data Mining) оскільки традиційні підходи (сигнатурні системи) не здатні ефективно виявляти нові типи атак [1]. Методи Data Mining дозволяють виявляти приховані закономірності, аномалії та поведінкові патерни у великих масивах даних, що є критично важливим для сучасних систем кіберзахисту [2].

Дослідження в сфері аналізу даних кіберзагроз показують, що найбільш ефективними все ж є гібридні підходи, які поєднують машинне навчання, глибинні нейронні мережі та статистичні методи [3]. Тому однією з актуальних задач є систематизація сучасних методів Data Mining для аналізу кіберзагроз та оцінка їх ефективності у практичних системах кібербезпеки. Вона містить в собі зокрема завдання класифікація методів Data Mining для кібербезпеки, визначення їх переваг та недоліків, аналіз сучасних напрямків розвитку.

Одним із базових методів Data Mining, що використовується для виявлення відомих атак є класифікація [4]. Основними алгоритмами якого є: Decision Trees, Random Forest, Support Vector Machine та Neural Networks, що ефективні при наявності розмічених даних та широко застосовуються у IDS-системах. Методи кластеризації дозволяють виявляти нові та невідомі атаки без використання навчальних вибірок. Основні алгоритмами цих методів є K-means та DBSCAN. Вони використовуються для групування мережевих подій та виявлення відхилень від нормальної поведінки. Метод виявлення аномалій є ключовим напрямом сучасної кібербезпеки, коли аномалія визначається як відхилення від нормальної поведінки системи. Основними підходами в цьому напрямку є: статистичні методи, машинне навчання, deep learning (autoencoders). Методи глибинного навчання Deep Learning дозволяють аналізувати складні нелінійні залежності у великих даних. Сучасні моделі глибинного навчання досягають високої точності (>97%) при виявленні атак. Гібридні моделі, що поєднують: сигнатурний аналіз, аномалійний аналіз та машинне навчання забезпечують більш високу точність та адаптивність систем IDS. Ключовим застосуванням Data Mining є системи виявлення вторгнень (IDS). Вони виконують задачі моніторингу мережевого трафіку, аналізу поведінки та виявлення атак, вони також базуються на AI/ML і здатні виявляти складні кіберзагрози у реальному часі. Після ретельного аналізу методів Data Mining, що використовуються у кібербезпеці отримано такі результати (табл. 1):

Таблиця 1
Аналіз методів Data Mining

№ п/п	Назва методу	Переваги	Недоліки
1	Класифікація	Висока точність	Потребує розмічених даних
2	Кластеризація	Виявлення нових атак	Низька інтерпретованість
3	Аномалії	Ефективні для zero-day	Багато хибних спрацювань
4	Deep Learning	Висока точність	Великі обчислювальні витрати
5	Гібридні	Найкраща ефективність	Складність реалізації

Таким чином нами виявлено, що методи Data Mining є одним із ключових інструментів сучасної кібербезпеки. Найбільш перспективними з них є виявлення аномалій, deep learning та гібридні IDS.

Список літератури

1. Смірнова Т.В., Усік П.С., Лисенко І.А., Буравченко К.О., Смірнов О.А. «Методологія підтримки технологічних процесів у критичній інфраструктурі з забезпеченням безпеки інформації на основі хмарних технологій». Безпека інформації. 2025. Том 31 № 1. С. 49-60.

2. Лисенко І.А., Минайленко Р.М., Смірнов С.А., Буравченко К.О., Якименко Н.М., Смірнов О.А. «Дослідження інструментів штучного інтелекту для інтелектуального аналізу даних». Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2025, № 3(31), С. 227–241.

3. Kawsar S. Using Data Mining as a Tool to Enhance Threat Detection and Response. IJETCSIT [Internet]. 2025 Oct. 10 [cited 2026 Apr. 5];:149-53.

4. Kuznetsov, O., Lysenko, I., Prokopovych-Tkachenko, D., Ulianovska, Y., Bushkov, V. «Dynamic Trust Evaluation and Resilience Assessment in Edge Computing Networks». International Journal of Computing, 2025, 24(2), pp. 223–232

УДК 004.056, 004.75, 621.9

О.В. Лисенко, І.А. Лисенко, В.А. Резніченко
ac74@i.ua, *min_max@i.ua*, *upsbilly@meta.ua*
Центральноукраїнський національний технічний університет, Кропивницький

ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ СИСТЕМ ДИСТАНЦІЙНОГО ДІАГНОСТУВАННЯ ОБРОБНИХ ЦЕНТРІВ FANUC

Сучасний розвиток концепції Industry 4.0 супроводжується глибокою інтеграцією обробних центрів з числовим програмним керуванням у цифрові виробничі середовища, що передбачає широке використання систем дистанційного моніторингу та діагностування [1, 2]. Застосування таких систем дозволяє підвищити ефективність експлуатації обладнання, скоротити час простоїв та забезпечити планування обслуговування. Водночас підключення CNC-систем до корпоративних мереж і хмарних сервісів формує нові вектори кіберзагроз, що зумовлює необхідність розробки комплексних підходів до забезпечення інформаційної безпеки [3, 4].

Типова архітектура систем дистанційного діагностування обробних центрів на базі FANUC включає інтеграцію CNC-контролера, програмованого логічного контролера, інтерфейсу оператора та периферійних сенсорів із edge-шлюзом збору та попередньої обробки даних [5]. Передача інформації до віддалених сервісів здійснюється через захищені канали зв'язку, зокрема VPN-з'єднання, із подальшою обробкою у хмарних платформах моніторингу. У такій архітектурі найбільш цінною інформацією є керуючі програми, параметри обробки, діагностичні дані та облікові дані доступу, компрометація яких може призвести до суттєвих виробничих і економічних втрат.

Для формалізації загроз було застосовано методологію STRIDE [6], яка дозволяє класифікувати потенційні атаки за шістьма основними категоріями. Зокрема, підміна або розширення прав користувача може реалізовуватися через компрометацію облікових даних віддаленого доступу, тоді як модифікація даних проявляється у несанкціонованій зміні параметрів ЧПК або керуючих програм. Відсутність належного протоколювання створює передумови для відмови від виконаних дій, а витік інформації може стосуватися технологічних режимів обробки та конструктивних особливостей виробів. Окрему небезпеку становлять атаки типу відмови в обслуговуванні, що порушують доступність систем моніторингу, а також підвищення привілеїв, яке надає зловмиснику повний контроль над функціонуванням обладнання. Узагальнену модель загроз наведено на рис. 1.

Рис. 1. Модель загроз для систем дистанційного діагностування обробних центрів (STRIDE та MITRE ATT&CK for ICS)

Згідно з рис. 1, основні загрози охоплюють порушення цілісності, конфіденційності та доступності системи. Подальша деталізація загроз здійснюється із використанням бази знань MITRE ATT&CK for ICS, що дозволяє співставити типові сценарії атак із конкретними техніками. Зокрема, експлуатація віддалених сервісів і використання нелегітимних облікових записів відкривають можливості для проникнення в систему, тоді як атаки типу «людина посередині» спрямовані на перехоплення та модифікацію переданих даних. Значну загрозу становлять також дії, пов'язані зі зміною параметрів функціонування системи та завантаженням модифікованих

керуючих програм, що безпосередньо впливає на технологічний процес. Додатково враховуються сценарії відмови в обслуговуванні, які можуть призвести до зупинки виробництва.

Аналіз отриманих результатів показує, що ефективне забезпечення інформаційної безпеки систем дистанційного діагностування повинно базуватися на поєднанні організаційних та технічних заходів. Зокрема, важливу роль відіграє сегментація мережі із виділенням окремих зон для виробничих та корпоративних ресурсів, використання захищених каналів зв'язку з багатофакторною автентифікацією, а також впровадження механізмів контролю цілісності керуючих програм. Не менш важливими є системи виявлення вторгнень, централізоване логування подій та реалізація принципу мінімальних прав доступу, що дозволяє обмежити можливості зловмисника у разі компрометації окремих компонентів системи. Розроблений підхід узгоджується з вимогами міжнародних стандартів інформаційної безпеки та промислових систем керування. Архітектуру захищеної системи дистанційного діагностування обробних центрів наведено на рис. 2.

Рис. 2. Архітектура захищеної системи дистанційного діагностування обробних центрів

Таким чином, запропонована модель загроз дозволяє систематизувати основні ризики інформаційної безпеки в системах дистанційного діагностування обробних центрів та створює теоретичну основу для розроблення ефективних засобів захисту. Практичне значення роботи полягає у можливості застосування отриманих результатів для підвищення рівня кіберзахищеності сучасних виробничих систем, що функціонують у середовищі Industry 4.0.

Список літератури

1. Лисенко О. В., Лисенко І. А. Метод забезпечення захисту передачі інформації систем дистанційного контролю та діагностики обробних центрів // Development strategies for modern education and science : Materials of the VI International Research and Practical Internet Conference (February 23-25, 2025) / за заг. ред. Ph.D Serhii Onyshchenko. – Zdar nad Sazavou (Czech Republic) : DEL c.z., 2025. – P. 29–31. – Режим доступу: https://dspace.kntu.kr.ua/handle/123456789/16183 (дата звернення: 06.04.2026).

2. Lee J., Bagheri B., Kao H.-A. A cyber-physical systems architecture for Industry 4.0-based manufacturing systems // Manufacturing Letters. – 2015. – Vol. 3. – P. 18-23. DOI: https://doi.org/10.1016/j.mfglet.2014.12.001

3. Nankya M., et al. Securing industrial control systems: components, cyber threats and challenges // Sensors. – 2023. – Vol. 23. – DOI: https://doi.org/10.3390/s23218840

4. Kasprzyczak L. Cybersecurity requirements for industrial machine control systems // Applied Sciences. – 2025. – Vol. 15, No. 3. – DOI: https://doi.org/10.3390/app15031267

5. Fanuc. CNC Systems and Industrial IoT Solutions. – Режим доступу: https://www.fanuc.eu (дата звернення: 06.04.2026).

6. OWASP Foundation. Threat Modeling Process. – Режим доступу: https://owasp.org/www-community/Threat_Modeling_Process (дата звернення: 06.04.2026).

УДК 004.056.53 (045)

С.Ю. Лисюк, *здобувач кафедри Кібербезпеки[1] першого (бакалаврського) рівня вищої освіти, другий курс,*
А.В. Ільєнко, *канд. техн. наук, доцент, завідувач кафедри* Кібербезпеки[1],
О.В. Дубчак, *ст. викладач кафедри Кібербезпеки[1]*
8481263@stud.kai.edu.ua , anna.ilienko@npp.kai.edu.ua, 3915922@npp.kai.edu.ua
[1]Державний університет «Київський авіаційний інститут», Київ, Україна

OSINT У СТРУКТУРІ СУЧАСНИХ КІБЕРАТАК: КЛАСИФІКАЦІЯ, РИЗИКИ ТА МЕТОДИ ПРОТИДІЇ

У сучасному інформаційному суспільстві дані є ключовим стратегічним ресурсом. Розвиток цифрових технологій призвів до того, що значна частина інформації про людей та організації опинилась у відкритому доступі. Саме цим користуються зловмисники, які ще до початку безпосередньої атаки ретельно збирають та аналізують публічні відомості про свою ціль.

OSINT (Open Source Intelligence — розвідка на основі відкритих джерел) є методологією, що охоплює систематичний збір, обробку та аналіз інформації з будь-яких загальнодоступних джерел: соціальних мереж, вебресурсів, технічних баз даних, державних реєстрів, медіа, супутникових знімків тощо. Ця методологія не є чимось суто «хакерським» — вона широко застосовується журналістами, аналітиками, правоохоронцями та фахівцями з кібербезпеки[1]. Проте ті самі інструменти й підходи активно використовуються кіберзлочинцями для підготовки фішингових кампаній, атак соціальної інженерії та складних APT-операцій[2].

Дослідники зазначають, що зростання кількості кібератак прямо пов'язане з удосконаленням методів попередньої розвідки: зловмисник, що має докладний профіль цілі, підвищує шанси на успіх атаки в рази[3]. Водночас більшість організацій досі недооцінює обсяг інформації, яку вони самі публікують у відкритому доступі через сайти, соціальні мережі та публічні документи. Це робить тему виявлення та протидії OSINT-розвідці надзвичайно актуальною як для корпоративного, так і для державного сектору.

Метою дослідження є системний аналіз використання OSINT-інструментів у процесі підготовки кібератак та обґрунтування підходів до мінімізації ризиків витоку інформації з відкритих джерел. Для досягнення поставленої мети визначено такі завдання: розкрити сутність OSINT та визначити його роль у циклі кібератаки; класифікувати джерела відкритої інформації та інструменти їх збору; проаналізувати основні сценарії використання OSINT зловмисниками; визначити методи виявлення OSINT-активності щодо організації; обґрунтувати рекомендації щодо протидії та мінімізації ризиків витоку інформації

Об'єктом дослідження є процес збору та використання інформації з відкритих джерел у контексті підготовки кібератак. Предметом дослідження — методи та інструменти OSINT-розвідки, а також технічні й організаційні засоби протидії витоку інформації.

Для досягнення мети застосовано: системний аналіз для класифікації типів OSINT та відповідних інструментів; порівняльний аналіз для оцінки функціоналу та сфер застосування конкретних програмних рішень (Maltego, Shodan, SpiderFoot, TheHarvester, Recon-ng, Paliscope, Intelligence X); метод синтезу — для розробки рекомендацій щодо контр-OSINT заходів на основі аналізу наукової літератури, технічної документації та відкритих звітів з кібербезпеки.

OSINT у циклі кібератаки. Згідно з моделлю Cyber Kill Chain, будь-яка цілеспрямована атака починається з фази розвідки. У фреймворку MITRE ATT&CK ця фаза (Reconnaissance) включає пасивний збір інформації без прямого контакту з ціллю. OSINT є основним інструментом на цьому етапі: зловмисник формує профіль цілі, виявляє слабкі місця та обирає оптимальний сценарій атаки ще до будь-яких активних дій. Збір інформації OSINT поділяється на три методи: пасивний (використання публічних ресурсів без взаємодії з ціллю), напівпасивний (обмежений трафік на цільовий сервер для отримання загальних відомостей) та активний (безпосередня взаємодія з системою, прозора для власника)[4].

Класифікація джерел і типів OSINT. За характером джерел розрізняють: засоби масової інформації (преса, телебачення, онлайн-видання); Інтернет-ресурси (соціальні мережі, блоги, форуми, YouTube); державні дані (урядові звіти, реєстри, прес-релізи, офіційні сайти); спостереження (радіомоніторинг та супутникові знімки). За типом аналізу виокремлюють: GeoINT — геопросторова розвідка (аналіз фото з EXIF-даними, GPS-координати, супутникові знімки); SOCMINT — аналіз публікацій у соціальних мережах; TECHINT — технічна розвідка (WHOIS, метадані зображень, аналіз SSL-сертифікатів)[5].

Сценарії використання OSINT зловмисниками. Корпоративний OSINT передбачає збір даних про доменну інфраструктуру, піддомени, відкриті порти, email-адреси співробітників та організаційну структуру компанії. Персональний OSINT — профілювання конкретної особи через соціальні мережі, геолокаційні дані та фотоаналіз. На основі зібраних даних реалізуються такі вектори атак: цільовий фішинг (Spear Phishing) — персоналізовані листи з реальними даними жертви; соціальна інженерія — маніпуляція через видавання себе за довірену особу або колегу; APT-кампанії (Advanced Persistent Threat) — тривале приховане проникнення з попередньою багатоетапною розвідкою. Зловмисники також активно застосовують Google Dorking — пошук

прихованої інформації через розширені оператори Google (filetype:, inurl:, intitle:), що дозволяє знаходити конфіденційні документи, відкриті бази даних та незахищені сторінки адміністрування[6].

Огляд ключових OSINT-інструментів. Maltego — графічний інструмент для аналізу зв'язків між людьми, доменами, IP-адресами та організаціями. Дозволяє будувати граф із до 1 млн об'єктів і автоматизувати пошук у 58+ джерелах даних. Shodan — «пошукова система для IoT», що індексує підключені до інтернету пристрої (вебкамери, маршрутизатори, промислові системи SCADA), виявляє відкриті порти та вразливості. TheHarvester збирає email-адреси, піддомени та IP-адреси за заданим доменом із пошукових систем та DNS-баз. SpiderFoot автоматизує збір даних із понад 100 публічних джерел і формує детальний звіт по об'єкту пошуку. Recon-ng — Python-фреймворк для розвідки, вбудований у Kali Linux, що підтримує роботу з базами даних та API і автоматизує трудомісткі OSINT-операції. PhoneInfoga дозволяє аналізувати телефонні номери, встановлювати країну, оператора та тип лінії, а потім проводити повноцінну OSINT-розвідку за номером. Intelligence X індексує витоки даних, документи та форуми даркнету, надає доступ до видалених матеріалів і забезпечує анонімність дослідника. Paliscope орієнтований на цифрові розслідування та судову експертизу — інтегрує штучний інтелект для прискорення обробки великих масивів даних і формує структуровані звіти[7,8]. Порівняльний аналіз інструментів наведено у таблиці 1.

Таблиця 1

Порівняльний аналіз ключових OSINT-інструментів

Інструмент	Призначення	Ключові можливості
Maltego	Аналіз зв'язків	Граф до 1 млн об'єктів, 58+ джерел
Shodan	Пошук пристроїв в мережі	IoT, SCADA, відкриті порти, вразливості
TheHarvester	Збір даних про домен	Email, піддомени, IP-адреси
SpiderFoot	Автоматизований збір	100+ джерел, звіт по об'єкту
Recon-ng	Розвідувальний фреймворк	API, бази даних, автоматизація
PhoneInfoga	Розвідка за телефоном	Оператор, країна, тип лінії
Intelligence X	Пошук у витоках і даркнеті	Видалені матеріали, анонімність
Paliscope	Цифрові розслідування	ШІ-обробка даних, структуровані звіти

Практичне застосування OSINT. У кібербезпеці OSINT застосовують для раннього виявлення загроз (моніторинг форумів даркнету та соціальних мереж на предмет обговорення вразливостей цілі), виявлення витоків даних та тестування на проникнення. У журналістиці — для розслідувань на основі державних реєстрів, судових рішень та тендерних майданчиків; побудови графів зв'язків між політиками та бізнесменами; верифікації фото- та відеоматеріалів через аналіз метаданих. У геополітичному аналізі — для моніторингу військових переміщень через супутникові знімки, відстеження пропагандистських наративів, верифікації подій у зонах конфліктів[9].

Методи виявлення OSINT-активності проти організації. Для виявлення того, що організація стала об'єктом OSINT-розвідки, застосовують кілька підходів. Моніторинг згадок бренду (Brand Monitoring) — відстеження публікацій про компанію, її співробітників та продукти в соціальних мережах, форумах та ЗМІ. Honeytokens та canary-пастки — підроблені облікові дані, документи або DNS-записи, звернення до яких свідчить про несанкціоноване сканування. Платформи Threat Intelligence (MISP, OpenCTI) — збір та аналіз даних про загрози для розуміння методів і дій зловмисників. Моніторинг даркнету — відстеження підпільних форумів та маркетплейсів на предмет появи корпоративних даних у відкритому продажу.

Технологічні засоби захисту та контр-OSINT. Counter-OSINT — це комплекс методів і стратегій, спрямованих на захист інформації від збору зловмисниками через відкриті джерела. Технічні засоби включають: шифрування даних (симетричні та асиметричні алгоритми AES, RSA) для захисту інформації під час передачі та зберігання; VPN та захищені канали зв'язку для приховування реальної IP-адреси; DLP-системи (Data Loss Prevention) для автоматичного блокування несанкціонованої передачі даних; SIEM-платформи для збору, аналізу та кореляції подій безпеки[10]. Організаційні заходи передбачають: формування чітких політик щодо публікації інформації у відкритому доступі; обмеження надлишкових публічних відомостей на корпоративних сайтах; регулярний аудит цифрового сліду організації та її ключових співробітників. Освітні заходи охоплюють: навчання персоналу основам кібергігієни та розпізнаванню фішингових атак; підвищення обізнаності щодо ризиків публікації службової інформації в соціальних мережах; проведення навчань з соціальної інженерії для перевірки готовності команди[11].

Висновки. OSINT є потужним двостороннім інструментом — критично важливим як для фахівців з кібербезпеки, так і для зловмисників. Широка доступність інструментів (Maltego, Shodan, SpiderFoot, Recon-ng та ін.) у поєднанні з величезним обсягом публічної інформації робить OSINT-розвідку невід'ємною складовою

підготовки сучасних кібератак. Ефективна протидія цим методам вимагає комплексного підходу, що поєднує технічні засоби захисту (DLP, SIEM, шифрування), організаційні політики мінімізації цифрового сліду та систематичне підвищення рівня обізнаності персоналу щодо ризиків публічного поширення інформації.

Список літератури

1. OSINT – розвідка відкритих джерел. URL: https://darkguard.com.ua/osint-rozvidka-vidkritih-dzherel-v-darkguard/

2. What Is an Advanced Persistent Threat (APT)? URL: https://www.cisco.com/site/us/en/learn/topics/security/what-is-an-advanced-persistent-threat-apt.html

3. Як працює OSINT-розвідка та чому небезпечно публікувати інформацію в інтернеті. URL: https://fakty.com.ua/ua/ukraine/suspilstvo/20220429-yak-praczyuye-osint-rozvidka-ta-chomu-nebezpechno-publikuvaty-informacziyu-v-internet/https://fakty.com.ua/ua/ukraine/suspilstvo/20220429-yak-praczyuye-osint-rozvidka-ta-chomu-nebezpechno-publikuvaty-informacziyu-v-internet/

4. OSINT Open Source Intelligence. Інструменти та методи: навчальний посібник / Користін О., Демедюк С., Ісмайлов К., Ланде Д. та ін., за заг. ред. Користіна О.Є., Демедюка С.В. – Київ: 7БЦ, 2025. 60 с. URL: https://aord.com.ua/cms/uploads/OSINT_Open_Source_Intelligence_Instrumenti_ta_metodi_31913d17f8.pdf

5. SOCMINT, GEOINT, COMINT: three sub-disciplines of OSINT explained. URL: https://incyber.org/en/article/socmint-geoint-comint-three-sub-disciplines-of-osint-explained

6. OSINT-інструменти 2025: Топ-10 рішень для збору та аналізу даних. URL: https://softlist.com.ua/ua/news/osint-instrumenty-2025-top-10-resheniy-dlya-sbora-i-analiza-dannyh

7. OSINT: технологія збору та аналізу даних з відкритих джерел. URL: https://softlist.com.ua/ua/news/osint-tekhnologiya-sbora-i-analiza-dannyh-iz-otkrytyh-istochnikov

8. Топ-10 кращих інструментів OSINT для розвідки з відкритим вихідним кодом. URL: https://softlist.com.ua/ua/news/top-10-luchshykh-ynstrumentov-osint-dlia-razvedki-s-otkrytym-ishodnym-kodom

9. Техніки OSINT і GEOINT з виявленням вежі 5G. URL: https://hackyourmom.com/kibervijna/tehniky-osint-i-geoint-z-vyyavlennyam-vezhi-5g/

10. Дослідження існуючих засобів та підходів до проведення osint в контексті інформаційної безпеки особи та держави. URL: https://science.lpnu.ua/sites/default/files/journal-paper/2025/may/38968/k250473-145-161_0.pdf

11. Захист інформації в комп'ютерних системах та мережах. Частина II: підручник / Ю.В. Костюк, П.М. Складанний. – Київ : Київський столичний університет імені Бориса Грінченка, 2026. – 386 с. URL: https://elibrary.kubg.edu.ua/id/eprint/56333/1/ZIKSM_part_2_2026_FITM.pdf

УДК 004.738.5:004.056:159.9

О.О. Попілевич, *студент 4 курсу*
Науковий керівник: С.В. Науменко, *викладач кафедри інформаційних технологій*
popilevych.oleksandr1122@vu.cdu.edu.ua
Черкаський національний університет імені Богдана Хмельницького, Черкаси

ДИЛЕМА АНОНІМНОСТІ НА ЦИФРОВИХ ПЛАТФОРМАХ ПСИХОЛОГІЧНОЇ ПІДТРИМКИ: АСПЕКТИ ІНФОРМАЦІЙНО-ПСИХОЛОГІЧНОЇ БЕЗПЕКИ

Стигматизація психічних розладів залишається одним із найбільш стійких бар'єрів, що перешкоджає людям звертатися по допомогу. Систематичний огляд 144 досліджень за участю понад 90 тисяч респондентів показав, що стигма має помірний негативний вплив на звернення по допомогу (d = −0,27), причому побоювання щодо розкриття інформації про своє захворювання є найпоширенішим бар'єром, пов'язаним зі стигмою [1]. Особливо вразливими до цього ефекту виявилися молоді люди, чоловіки та представники етнічних меншин. Для України, де 77 % населення повідомляють про переживання стресу, а 78 % ніколи не зверталися до психолога [2], проблема стигми набуває критичного значення.

У відповідь на цю проблему активно розвиваються цифрові платформи психологічної підтримки, які пропонують анонімність як ключову перевагу. Анонімність у цьому контексті виконує функцію психологічного захисту: вона дозволяє користувачам обійти механізми самоцензури, зумовлені страхом соціального осуду. Дослідження Qian та Wahl (2013) на вибірці з 114 авторів блогів про здоров'я продемонструвало, що анонімність стратегічно використовується людьми, які відчувають сором через свій стан, і саме вона модерує зв'язок між соромом та готовністю до саморозкриття: серед анонімних блогерів сором позитивно корелював із глибиною розкриття інформації [3]. Тобто анонімність не просто знімає бар'єр — вона перетворює його на ресурс для терапевтичної комунікації.

Однак анонімність на цифрових платформах має і зворотний бік, що становить серйозний виклик для інформаційно-психологічної безпеки. Реалістичний синтез Sheridan et al. (2024), присвячений аналізу механізмів впливу онлайн-форумів peer-підтримки, виявив подвійну природу цього явища. З одного боку, анонімність через механізм розгальмування (disinhibition effect) сприяє саморозкриттю та формуванню відчуття спільності. З іншого — та сама розгальмованість може провокувати антисоціальну поведінку: булінг, поширення деструктивного контенту та маніпуляції [4].

Таким чином, формується протиріччя між двома вимогами інформаційно-психологічної безпеки: з одного боку, анонімність необхідна для подолання стигми та забезпечення психологічної безпеки користувача при саморозкритті; з іншого — вона створює умови для порушення безпеки інших учасників спільноти. Вирішення цієї дилеми потребує багаторівневого підходу: архітектурного (технічна анонімність без збору персональних даних), комунітарного (правила спільноти та системи скарг) та інтелектуального (автоматизоване виявлення кризового контенту).

Отже, анонімність на цифрових платформах психологічної підтримки є не стільки технічною характеристикою, скільки інструментом інформаційно-психологічної безпеки, ефективність якого залежить від контексту застосування. Для забезпечення безпечного цифрового середовища необхідний баланс між захистом приватності користувача та захистом спільноти від деструктивної поведінки. Подальші дослідження мають бути спрямовані на розробку адаптивних моделей модерації, що враховують специфіку анонімних peer-to-peer спільнот в умовах масштабних психосоціальних криз, зокрема воєнних конфліктів.

Список літератури

1. Clement S., Schauman O., Graham T. et al. What is the impact of mental health-related stigma on help-seeking? A systematic review of quantitative and qualitative studies. Psychological Medicine. 2015. Vol. 45, No. 1. P. 11–27.

2. Психічне здоров'я та ставлення українців до психологічної допомоги під час війни : 3 хвиля / Gradus Research. Київ, 2024. URL: https://gradus.app/uk/open-reports/mental-health-and-attitudes-ukrainians-towards-psychological-assistance-during-war/ (дата звернення: 05.04.2026).

3. Qian H., Wahl S. T. The implications of stigma and anonymity for self-disclosure in health blogs. Health Communication. 2013. Vol. 28, No. 1. P. 59–68.

4. Sheridan G. et al. Understanding the Impacts of Online Mental Health Peer Support Forums: Realist Synthesis. JMIR Mental Health. 2024. Vol. 11. e55750.

УДК 004.056.5

Є.А. Якимчук[1], Я.В. Марченко[1],
[1]асистенти кафедри Кібербезпеки[1]
yevhenii.iakymchuk@npp.kai.edu.ua, yaroslav.marchenko@npp.kai.edu.ua
Державний університет «Київський авіаційний інститут», Київ, Україна

КЛАСИФІКАЦІЯ ВЕБЗАГРОЗ У СИСТЕМАХ З ІНТЕЛЕКТУАЛЬНИМИ ПОМІЧНИКАМИ ТА ПІДХІД ДО ЇХ МОДЕЛЮВАННЯ

Інтеграція інтелектуальних помічників у сучасні вебсистеми суттєво змінює характер кіберзагроз. За оцінками галузевих аналітиків, частка вебдодатків з інтегрованими AI-компонентами стрімко зростає, і вже найближчими роками більшість корпоративних інформаційних систем використовуватиме інтелектуальних помічників як основний інтерфейс взаємодії з користувачем [3]. Якщо класичні вебвразливості - SQL-ін'єкції, міжсайтовий скриптинг, порушення автентифікації - виникають через синтаксичні помилки реалізації і систематизовані у загальновідомих стандартах [1, 2], то AI-орієнтовані загрози реалізуються на іншому рівні: через інтерпретацію намірів користувача, агрегацію даних із різних сервісів і автономну логіку прийняття рішень помічником. Ця відмінність є суттєвою - традиційні засоби тестування безпеки, зокрема DAST-сканери та WAF, орієнтовані на детерміновану поведінку компонентів і не здатні охопити семантичний рівень взаємодії, що робить AI-орієнтовані загрози структурно невидимими для стандартного пентесту та інструментів тестування API [2].

Ключова архітектурна відмінність систем з інтелектуальними помічниками полягає у появі недетермінованого посередника між наміром користувача і виконанням системних операцій. У класичній вебсистемі конкретна дія користувача породжує конкретний API-виклик із чітко визначеними параметрами, авторизація перевіряється на рівні цього виклику, а межі дозволеного явно вказані. Натомість інтелектуальний помічник самостійно визначає, до яких сервісів звернутися, які дані агрегувати і як інтерпретувати межі запиту - рішення, що раніше були жорстко визначені, тепер залежать від контексту і стану моделі [3, 7]. Саме це зміщення є джерелом нового класу загроз, які не можуть бути адекватно описані існуючими фреймворками моделювання, зокрема STRIDE [5].

На основі аналізу галузевих досліджень [3, 4, 6, 7] виокремлено п'ять категорій загроз, характерних для вебсистем з інтелектуальними помічниками. Маніпуляція намірами користувача (prompt injection) передбачає вбудовування у запит або оброблювані дані інструкцій, що змінюють логіку виконання помічника: зловмисник маніпулює не синтаксисом запиту, а його семантикою. Особливою формою такої атаки є indirect prompt injection - впровадження шкідливих інструкцій у зовнішні дані, що обробляються помічником (документи, вебсторінки, відповіді API), без безпосередньої участі користувача-зловмисника [7]. Витік інформації через агрегацію виникає, коли помічник об'єднує результати кількох технічно авторизованих викликів, розкриваючи інформацію, доступ до якої явно не надавався жодним окремим викликом - вразливість, що не існує в жодному окремому сервісі і не фіксується жодним логом як порушення. Некоректна робота моделі проявляється у впевненому наданні хибної інформації про стан ресурсів або права доступу без відображення помилки у системних логах, що унеможливлює автоматичне виявлення. Надмірні повноваження агента означають використання помічником наданого доступу поза межами наміру конкретного запиту - на відміну від класичного privilege escalation, жодне технічне обмеження при цьому не порушується. Порушення контексту доступу виникає через семантичну неоднозначність запиту, коли помічник обирає ширше трактування меж доступу, ніж передбачено політиками безпеки. Систематизацію технік атак на AI-компоненти наведено у каталозі MITRE ATLAS [7].

Для систематичного аналізу цих загроз запропоновано чотирирівневу модель, що розширює підхід STRIDE [5] шляхом явного включення інтелектуального помічника як окремого учасника з власною логікою прийняття рішень. Рівень взаємодії з користувачем охоплює загрози маніпуляції через природномовні запити, зокрема prompt injection. Рівень інтерпретації запиту є принципово новим - він не має відповідника у класичних вебсистемах і є основним джерелом контекстних вразливостей; його тестування вимагає спеціалізованих методів, оскільки поведінка є непередбачуваною і залежить від стану моделі. Рівень оркестрації сервісів охоплює ризики агрегаційного витоку та надмірних повноважень агента при паралельних викликах до незалежних backend-сервісів. Рівень доступу до ресурсів включає витоки через відсутність перехресного контролю доступу між сховищами даних, а також загрозу зараження пам'яті помічника (memory poisoning) - впровадження шкідливих інструкцій у довготривалу пам'ять агента, ефект якого може зберігатися між різними сесіями та користувачами.

Аналіз ефективності традиційних засобів підтверджує необхідність запропонованого підходу. За даними ENISA [3], стандартні інструменти тестування безпеки виявляють менше 25% загроз, пов'язаних із логікою роботи AI-компонентів. Behl et al. [4] підтверджують, що ручний пентест охоплює лише частину логічних вразливостей у AI-інтегрованих системах через відсутність стандартизованих методик тестування семантичного рівня. Chandrasekaran et al. [6] вказують на обмеженість автоматизованих сканерів щодо

виявлення загроз, що реалізуються через агрегацію даних. Це узгоджується з висновком про те, що чотири з п'яти виокремлених категорій загроз є невидимими для DAST-сканерів і WAF, а виявлення вимагає аналізу повної сесії помічника з урахуванням контексту виконання операцій.

З практичної точки зору, отримані результати вказують на необхідність доповнення існуючих стандартів безпечної розробки вимогами, специфічними для AI-інтеграцій: застосування принципу мінімальних привілеїв на рівні агента, обов'язкова перевірка та звуження контексту запиту перед ініціюванням сервісних викликів, логування намірів сесій помічника для подальшого аудиту, а також впровадження механізмів підтвердження для операцій [3, 5].

Отже, безпека вебсистем з інтелектуальними помічниками вимагає перенесення акценту з синтаксичного контролю вхідних даних на семантичний аналіз намірів і контексту виконання [1]. Запропонована класифікація п'яти категорій загроз і чотирирівнева модель аналізу можуть слугувати основою для розробки спеціалізованих методик тестування, доповнення існуючих стандартів безпечної розробки (OWASP ASVS, NIST SSDF) вимогами, специфічними для AI-інтеграцій, а також для подальших емпіричних досліджень на реальних системах з AI-помічниками.

Разом з тим, подальший розвиток підходів до моделювання загроз має враховувати динамічний характер поведінки інтелектуальних агентів, зокрема їх здатність адаптуватися до контексту взаємодії та змінювати стратегії обробки запитів. Це створює потребу у формуванні нових підходів до верифікації безпеки, що поєднують класичні методи контролю доступу з аналізом поведінкових та семантичних характеристик системи. Крім того, важливим напрямом є розробка інструментів автоматизованого тестування, здатних враховувати повний контекст сесії взаємодії з помічником, що дозволить підвищити ефективність виявлення складних логічних та контекстних вразливостей.

Окремої уваги заслуговує питання інтеграції запропонованого підходу у процеси безпечної розробки програмного забезпечення. Зокрема, доцільним є включення етапів аналізу поведінки інтелектуального помічника до життєвого циклу розробки (SDLC), а також адаптація існуючих практик threat modeling з урахуванням нових типів взаємодії. Використання запропонованої моделі дозволяє формалізувати аналіз ризиків на ранніх етапах проєктування системи та забезпечити більш системний підхід до виявлення потенційних вразливостей ще до їх реалізації у продуктивному середовищі.

Важливим напрямком подальших досліджень є також розробка моделей оцінювання ризиків для систем з інтелектуальними помічниками, які дозволяють визначати рівень впливу контекстних вразливостей. Це, у свою чергу, сприятиме побудові кращих стратегій захисту та пріоритезації заходів безпеки у складних розподілених системах. Практичне застосування запропонованої моделі також відкриває можливості для побудови адаптивних систем захисту, які здатні динамічно реагувати на зміну поведінки інтелектуального помічника та контексту взаємодії. Зокрема, впровадження механізмів політик безпеки, що враховують не лише роль користувача, але й інтерпретований намір запиту, дозволяє значно зменшити ризики неконтрольованого доступу до ресурсів. Такий підхід може бути реалізований через поєднання контекстно-орієнтованого контролю доступу (context-aware access control) та аналізу поведінки агентів у реальному часі.

Крім того, доцільним є використання підходів explainable AI для підвищення прозорості рішень, що приймаються інтелектуальними помічниками. Це дозволить не лише покращити довіру до системи, але й забезпечити можливість аудиту дій агента з точки зору безпеки. У поєднанні з механізмами журналювання та відстеження контексту виконання, це створює основу для більш ефективного розслідування інцидентів та виявлення складних багаторівневих атак.

Список літератури

1. OWASP Foundation. (2023). OWASP Top 10: The Ten Most Critical Web Application Security Risks. https://owasp.org/www-project-top-ten/

2. OWASP Foundation. (2023). OWASP API Security Top 10. https://owasp.org/www-project-api-security/

3. European Union Agency for Cybersecurity (ENISA). (2023). ENISA Threat Landscape for Artificial Intelligence. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

4. Behl, A., Behl, K., & Behl, D. (2022). Security implications of artificial intelligence in web applications. Journal of Cybersecurity and Privacy, 2(3), 512–528.

5. Shostack, A. (2014). Threat Modeling: Designing for Security. Wiley.

6. Chandrasekaran, M., Amirthalingam, P., & Palanisamy, V. (2021). Security threats and mitigation techniques in artificial intelligence systems. Future Internet, 13(8), 201.

7. MITRE Corporation. (2023). MITRE ATLAS: Adversarial Threat Landscape for Artificial Intelligence. https://atlas.mitre.org

УДК 004.056

К. Бегунець, *аспірант 1-го курсу,*
Науковий керівник: О. Висоцька, *к.т.н., доцент кафедри кібербезпеки.*
4817730@stud.kai.edu.ua
Державний університет «Київський авіаційний інститут», Київ

БЕЗПЕЧНА НЕВИКОНУВАНА ДЕОБФУСКАЦІЯ VBS/HTA-СЦЕНАРІЇВ НА ОСНОВІ ПРОМІЖНОГО ПОДАННЯ ТА SSA

Вступ.

У наукових працях переважають підходи до деобфускації виконуваного коду [3, 7], PowerShell-сценаріїв [2] та JavaScript-програм [4, 5, 6, 8]. Для VBScript наявні переважно засоби виявлення шкідливих сценаріїв, а не їх безпечної невиконуваної деобфускації [1]. Обфускація сценаріїв VBScript та HTA (HTML Application) приховує мережеві адреси, шляхи, команди запуску й імена об'єктів автоматизації, що ускладнює первинний статичний аналіз. Отже, у наявних дослідженнях недостатньо опрацьовано підходи до безпечної невиконуваної деобфускації VBS/HTA-сценаріїв. Тому розроблення безпечного невиконуваного прототипу деобфускації VBS/HTA-сценаріїв є актуальною задачею кібербезпеки.

Мета роботи – розробити дослідницький прототип мовою Python для безпечної невиконуваної деобфускації обфускованих файлів *.vbs і *.hta, який на виході формує деобфускований VBS, абстрактне синтаксичне дерево, проміжне подання та SSA-форму, а також дослідити здатність такого конвеєра відновлювати попередньо розмічені артефакти первинного статичного аналізу.

Матеріали та методи.

Матеріалами дослідження стали знешкоджені набори зразків обфускованих сценаріїв VBScript та HTA, підготовлені для відтворюваної перевірки, і сучасні наукові праці з деобфускації та аналізу сценарного коду [1, 2, 4, 6], а також узагальнювальні роботи з автоматичної деобфускації [3, 7]. Експериментальні набори зразків: 1) базовий набір – 7 модельних сценаріїв; 2) синтетичний набір – 8 узагальнених синтетичних сценаріїв; 3) похідний набір – 9 сценаріїв, побудованих за типовими мотивами обфускації; 4) об'єднаний набір – 24 сценарії; 5) розширений об'єднаний набір – 96 сценаріїв після лексичних мутацій. Очікуваним артефактом вважали наперед розмічений аналітично корисний рядок: мережеву адресу, шлях, ім'я об'єкта автоматизації COM (Component Object Model), команду запуску або інший змістовний індикатор у службових метаданих. Відновленим артефактом вважали артефакт, знайдений у нормалізованому виході після приведення hxxp(s) до http(s) та згортання подвоєних VBS-лапок. Принцип роботи прототипу складається з чотирьох етапів: 1) вилучення VBS із HTA; 2) побудова абстрактного синтаксичного дерева для підтримуваної підмножини VBScript; 3) локальні деобфускаційні перетворення і трансляція у графове проміжне подання з базовими блоками та явними переходами; 4) побудова форми єдиного статичного присвоювання(SSA), формування очищеного VBS і вилучення артефактів із підсумкового тексту. Задача розв'язується за рахунок згортання конкатенацій, Chr/ChrW, Replace, Split/Join, Left/Right/Mid, StrReverse, безпечного розгортання Execute/ExecuteGlobal, усунення самоприсвоєнь і мертвих присвоєнь.

Результати.

Запропонований прототип на вхід приймає обфускований VBS/HTA-сценарій, а на виході формує деобфускований VBS, абстрактне синтаксичне дерево, проміжне подання, SSA-форму та статистику відновлення артефактів. Експериментальне дослідження проводили у чотирьох станах: до оброблення, після рядкового контрольного варіанта, після побудови структурного подання та після повного конвеєра. Рядковий контрольний варіант є мінімальним набором правил оброблення рядків без побудови абстрактного синтаксичного дерева та проміжного подання. Додатково, без виконання, локально перевірено набір карантинних VBS-зразків із захищеного архіву. У записі виду 73/76 перше число позначає кількість відновлених артефактів, друге – загальну кількість попередньо розмічених артефактів. На об'єднаному наборі результат зростає з 41/76 до 73/76 після побудови структурного подання, а повний конвеєр доводить його до 76/76; на розширеному об'єднаному наборі повний конвеєр відновив 304/304 попередньо розмічених артефактів. Це показує, що основний приріст забезпечується не лише рядковими правилами, а насамперед структурним аналізом сценарію.

Таблиця 1

Результати відновлення артефактів на наборах зразків

Набір	К-сть зразків	К-сть артефактів	До оброблення	Після рядкового контролю	Після структурного подання	Після повного конвеєра
базовий	7	22	0/22	13/22	20/22	22/22
синтетичний	8	16	4/16	4/16	16/16	16/16
похідний	9	38	13/38	24/38	37/38	38/38
об'єднаний	24	76	17/76	41/76	73/76	76/76
розширений	96	304	44/304	164/304	292/304	304/304

Практичне значення.

У результаті використання прототипу аналітик отримує не лише відновлені рядки, а структурно нормалізований сценарій, придатний до ручного аналізу, побудови правил виявлення та швидкого відбору ознак для подальшого розбору інциденту. Наявність очищеного VBS, AST, проміжного подання та SSA-форми дає змогу зіставляти результати різних

етапів аналізу, відокремлювати корисні індикатори компрометації від шуму обфускації та зменшує навантаження на аналітика під час первинного розбору зразка. Такий вихід може бути використаний для документування інциденту, підготовки ознак виявлення та швидкого формування аналітичних звітів. Прототип може використовуватися як перший безпечний етап попереднього статичного аналізу тоді, коли виконання підозрілого сценарію небажане.

Обговорення результатів.

Запропонований конвеєр деобфускації є безпечним для первинного аналізу, оскільки не передбачає запуску досліджуваного зразка. На відміну від суто динамічних підходів, такий прототип не створює ризику ненавмисного виконання шкідливого навантаження у процесі попереднього розбору. Побудова проміжного подання з базовими блоками та явними переходами дає змогу працювати не лише з окремими рядками, а і з залежностями між присвоєннями, конкатенаціями, викликами стандартних функцій та керувальними переходами. SSA-представлення додатково спрощує відстеження ланцюгів присвоєння і точок злиття керування в прикладах із розгалуженнями. Саме тому структурний етап забезпечує основний приріст відновлення артефактів порівняно з суто рядковими правилами, а вихід сценарію стає придатнішим для ручного розбору, формування аналітичних ознак і подальшого порівняння між зразками.

Обмеження дослідження.

Прототип орієнтований на підтримувану підмножину VBScript і не охоплює всіх варіантів динамічного формування коду, непрямих викликів через COM-диспетчеризацію та складних взаємодій між VBS і вбудованим JavaScript у HTA. Крім того, у межах цієї роботи не проведено ізольованого кількісного порівняння внеску SSA, усунення повторних виразів і видалення мертвих присвоєнь. Тому наведені результати слід трактувати як підтвердження працездатності всього конвеєра деобфускації загалом, а не як окремо доведений ефект кожної його стадії.

Висновки.

У роботі підтверджено працездатність безпечного невиконуваного прототипу деобфускації VBS/HTA-сценаріїв для підтримуваної підмножини конструкцій. Результати проведеного експериментального дослідження показали повне відновлення попередньо розмічених артефактів на всіх підготовлених наборах, зокрема 76/76 на об'єднаному та 304/304 на розширеному об'єднаному наборі. Отже, запропонований прототип може слугувати основою безпечного інструмента первинного статичного аналізу для задач кібербезпеки. Подальший розвиток доцільно спрямувати на розширення підтримуваних конструкцій VBScript, точніше опрацювання непрямих викликів, підтримку змішаних VBS/JS-HTA-сценаріїв та окреме оцінювання внеску SSA-етапу в загальний результат деобфускації.

Список літератури

1. Stokes J. W., Agrawal R., McDonald G. Detection of Malicious VBScript Using Static and Dynamic Analysis with Recurrent Deep Learning. ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing. 2020. P. 2887–2891. DOI: 10.1109/ICASSP40776.2020.9054390.

2. Li R., Zhang C., Chai H., Ying L., Duan H., Tao J. PowerPeeler: A Precise and General Dynamic Deobfuscation Method for PowerShell Scripts. Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security. 2024. DOI: 10.1145/3658644.3670310.

3. Yadegari B., Johannesmeyer B., Whitely B., Debray S. A Generic Approach to Automatic Deobfuscation of Executable Code. 2015 IEEE Symposium on Security and Privacy. 2015. DOI: 10.1109/SP.2015.47.

4. Chen T., Li D., Zhang Y., Xie T. JSimpo: Structural Deobfuscation of JavaScript Programs. ACM Transactions on Software Engineering and Methodology. 2025. DOI: 10.1145/3714460.

5. AbdelKhalek M., Shosha A. JSDES: An Automated De-Obfuscation System for Malicious JavaScript. Proceedings of the 12th International Conference on Availability, Reliability and Security. 2017. DOI: 10.1145/3098954.3107009.

6. Xu W., Zhang F., Zhu S. JStill: Mostly Static Detection of Obfuscated Malicious JavaScript Code. Proceedings of the Third ACM Conference on Data and Application Security and Privacy. 2013. DOI: 10.1145/2435349.2435372.

7. Kochberger P., Schrittwieser S., Schweighofer S., Kieseberg P., Weippl E. SoK: Automatic Deobfuscation of Virtualization-Protected Applications. Proceedings of the 16th International Conference on Availability, Reliability and Security. 2021. DOI: 10.1145/3465481.3465772.

8. Blanc G., Ando R., Kadobayashi Y. Term-Rewriting Deobfuscation for Static Client-Side Scripting Malware Detection. 4th IFIP International Conference on New Technologies, Mobility and Security. 2011. DOI: 10.1109/NTMS.2011.5720649.

УДК 004.056:53, 004.75

Д.В. Макаренко[1], І.О. Розломій[2]
dimamakar188@gmail.com, *inna-roz@ukr.net*
[1]*Черкаський державний фаховий бізнес-коледж, Черкаси*
[2]*Черкаський державний технологічний університет, Черкаси*

МЕТОДИ ВИЯВЛЕННЯ ТА ЗАПОБІГАННЯ КІБЕРТЕРОРИСТИЧНИМ АТАКАМ

У сучасних умовах цифровізації суспільства інформаційні системи та комп'ютерні мережі стали невід'ємною частиною функціонування держави, економіки та соціальної сфери. Зростання залежності від інформаційних технологій супроводжується підвищенням рівня кіберзагроз, серед яких особливо небезпечними є кібертерористичні атаки. Вони можуть порушувати роботу критичної інфраструктури, спричиняти економічні збитки та викликати соціальну напругу. У зв'язку з цим актуальним завданням є дослідження сучасних методів виявлення та запобігання таким атакам.

Інтернет активно використовується кіберзлочинцями для здійснення атак на критичну інфраструктуру, зокрема аеропорти та комерційні авіалінії. У відкритому доступі, зокрема в даркнеті, поширюються інструменти та інструкції для кібератак. Сучасні атаки характеризуються здатністю змінювати свої сигнатури та код з метою уникнення виявлення системами виявлення вторгнень (IDS) [1].

За останні десятиліття Україна швидко розвивалася, особливо у сфері цифровізації, і ще до початку російського вторгнення у лютому 2022 року Україна стикалася з серйозними кібератаками [2].

Кібертерористичні атаки можуть реалізовуватися у різних формах і часто мають комплексний характер. Найпоширенішими є розподілені атаки відмови в обслуговуванні (DDoS), які спрямовані на перевантаження серверів і мережевих ресурсів, атаки типу «людина посередині», що дозволяють перехоплювати та змінювати передану інформацію, а також використання шкідливого програмного забезпечення, зокрема вірусів, троянів і програм-вимагачів. Значну небезпеку становлять фішингові атаки та методи соціальної інженерії, які експлуатують людський фактор і дозволяють отримувати конфіденційні дані без безпосереднього технічного втручання. Сучасні атаки можуть виконуватися віддалено, залишатися непомітними для користувача та одночасно вражати велику кількість систем.

Одним із базових інструментів виявлення кіберзагроз є системи виявлення вторгнень (IDS), які здійснюють моніторинг мережевого трафіку та аналіз подій. Вони дозволяють своєчасно виявляти підозрілу активність та потенційні атаки. Сигнатурний підхід, що використовується в IDS, базується на порівнянні даних із відомими шаблонами атак, що забезпечує високу точність при виявленні відомих загроз. Водночас аномальний підхід дозволяє ідентифікувати нові, раніше невідомі атаки шляхом аналізу відхилень від нормальної поведінки системи.

Розширенням функціональності IDS є системи запобігання вторгнень (IPS), які не лише виявляють загрози, але й автоматично реагують на них. Такі системи здатні блокувати шкідливий трафік, обмежувати доступ до ресурсів або ізолювати заражені вузли, що зменшує час реагування на інциденти та знижує потенційні втрати. Використання IPS є важливим елементом побудови активної системи захисту. Ефективність таких систем значною мірою залежить від правильного налаштування та регулярного оновлення баз даних загроз.

Сучасні підходи до виявлення кібертерористичних атак дедалі частіше базуються на використанні методів машинного навчання та штучного інтелекту. Такі технології дозволяють обробляти великі обсяги інформації, виявляти приховані закономірності та прогнозувати можливі загрози. Зокрема, алгоритми класифікації можуть визначати підозрілу активність користувачів, а методи кластеризації – виявляти аномалії у мережевому трафіку. Нейронні мережі підвищують точність виявлення складних і прихованих атак (табл. 1).

Таблиця 1
Порівняння сучасних методів виявлення та запобігання кібертерористичним атакам

№ п/п	Метод	Тип	Принцип роботи	Переваги	Недоліки	Приклади застосування
1	IDS (сигнатурний аналіз)	Виявлення	Порівняння трафіку з базою відомих атак	Висока точність для відомих загроз	Не виявляє нові атаки	Виявлення вірусів, DDoS

Продовження Таблиці 1

2	IDS (аномальний аналіз)	Виявлення	Аналіз відхилень від нормальної поведінки	Виявляє нові атаки	Хибні спрацювання	Нетиповий трафік, insider threats
3	IPS	Запобігання	Автоматичне блокування шкідливої активності	Швидка реакція	Може блокувати нормальний трафік	Захист серверів
4	SIEM	Виявлення	Збір і кореляція подій з різних джерел	Комплексне бачення безпеки	Висока складність і вартість	SOC-центри
5	Машинне навчання	Виявлення	Аналіз великих обсягів даних і поведінки	Виявлення складних атак	Потребує навчання моделей	Аналіз поведінки користувачів
6	Нейронні мережі	Виявлення	Глибокий аналіз шаблонів і аномалій	Висока точність	Ресурсоємність	Виявлення APT-атак
7	Антивірусне ПЗ	Запобігання	Виявлення та видалення шкідливого ПЗ	Простота використання	Обмежена ефективність проти нових загроз	Захист ПК
8	Міжмережеві екрани	Запобігання	Фільтрація мережевого трафіку	Контроль доступу	Не аналізує складні атаки	Захист мереж
9	Криптографічний захист	Запобігання	Шифрування даних	Захист конфіденційності	Не запобігає атакам напряму	Передача даних
10	Багатофакторна автентифікація	Запобігання	Підтвердження особи кількома способами	Високий рівень безпеки	Незручність для користувача	Банківські системи
11	Резервне копіювання	Відновлення	Створення копій даних	Швидке відновлення	Не запобігає атакам	Захист від ransomware

Важливу роль у забезпеченні кібербезпеки відіграють системи управління подіями інформаційної безпеки (SIEM), які забезпечують централізований збір, обробку та кореляцію даних з різних джерел. Це дає змогу швидко виявляти загрози та реагувати на інциденти. Завдяки SIEM можна виявляти складні атаки, що реалізуються одночасно на різних рівнях інфраструктури.

Серед методів запобігання кібертерористичним атакам важливу роль відіграє криптографічний захист інформації. Використання сучасних алгоритмів шифрування забезпечує конфіденційність та цілісність даних навіть у разі їх перехоплення.

Суттєвим фактором забезпечення кібербезпеки є людський аспект. Багато кіберінцидентів виникає через помилки користувачів або недотримання правил безпеки.

Ефективне виявлення та запобігання кібертерористичним атакам можливе лише за комплексного підходу. Він передбачає використання сучасних систем моніторингу, засобів захисту, криптографічних методів і підвищення кіберграмотності користувачів. Це забезпечує надійний захист інформаційних систем і стабільну роботу критичної інфраструктури.

Список літератури

1. Lazaro Florido-Benitez (2024). The types of hackers and cyberattacks in the aviation industry.

2. Fyshchuk, I., Noesgaard, M. S., & Nielsen, J. A. (2024). Managing cyberattacks in wartime: The case of Ukraine. Public administration review.

УДК 004.056.5

С.В. Чернов[1], В.М. Чешун[1], Д.В. Чешун[2], Д.А. Олексюк[2]
odth.vip@gmail.com, cheshunvn@khmnu.edu.ua, dmytro.cheshun@gmail.com, oleksuk.dima@gmail.com
Хмельницький національний університет, Хмельницький
[2]Хмельницький фаховий економіко-технологічний коледж, Хмельницький

МОДУЛЬНА СИСТЕМА ПОШУКУ КРИТИЧНИХ ТОЧОК КОМПРОМЕТАЦІЇ ЗА ДОПОМОГОЮ GOOGLE DORKING

Постановка проблеми. Глобальна цифровізація сучасного соціуму зумовлює інтенсивну інтеграцію інформаційно-комунікаційних технологій у всі сфери життєдіяльності, що, поряд із розширенням операційних можливостей, детермінує виникнення нових критичних загроз у площині кібербезпеки. Динаміка зростання кількості інцидентів, пов'язаних із експлуатацією системних уразливостей та несанкціонованим доступом до конфіденційної інформації, набуває загрозливих масштабів. Згідно з релевантними аналітичними звітами, щорічно верифікуються мільйони кейсів порушення цілісності та конфіденційності даних, що становить пряму загрозу як для приватного сектора, так і для об'єктів критичної інфраструктури.

Аналіз останніх досліджень і публікацій. Фінансові втрати є одними із найвідчутніших наслідків витоку даних, що створюють серйозні виклики для бізнесу. Як свідчить звіт IBM Cost of Data Breach Report 2023 [1], середня вартість одного інциденту з витоку даних досягла 4,45 мільйона доларів США, що на 2,3% перевищує показники попереднього року [2]. Дані втрати охоплюють широкий спектр витрат: компенсації постраждалим клієнтам, витрати на реагування, створення спеціалізованих команд для розслідування, оновлення систем кіберзахисту, впровадження багатофакторної автентифікації та модернізацію інфраструктури безпеки. Додатково, організації несуть витрати на юридичні послуги та штрафи за недотримання нормативних актів, таких як GDPR [3] та CCPA [4]. У 2023 році ірландська Комісія із захисту даних наклала на компанію Meta штраф у розмірі 1,2 мільярда євро через неналежний обіг персональних даних [5]. Іншою знаковою подією став витік даних компанії Yahoo у 2013 році, виявлений у 2016-му, який призвів до зменшення вартості її продажу на 350 мільйонів доларів [6]. До того ж витрати на інцидент-менеджмент часто складають до 30% загальної суми втрат, особливо для бізнесів із високою залежністю від цифрових технологій. Наслідки не обмежуються лише фінансовими показниками – компанії також змушені зупиняти бізнес-процеси, проходити повторні аудити та переглядати політики безпеки, що ще більше погіршує ситуацію.

Водночас не менш значущими є репутаційні втрати, які можуть тривалий час завдавати шкоди бізнесу незалежно від його розміру чи галузі. Дослідження показують, що до третини клієнтів у секторах роздрібної торгівлі, фінансів та охорони здоров'я припиняють співпрацю з компаніями, які зазнали витоку даних. Це відображає низький рівень довіри споживачів до організацій, які не змогли забезпечити захист їхньої конфіденційної інформації. У результаті компанії втрачають не тільки доходи, але й значну частину клієнтської бази. Окрім втрати довіри клієнтів, репутаційні наслідки ускладнюють залучення інвесторів та кваліфікованих кадрів. У потенційних інвесторів виникають сумніви щодо фінансової стабільності бізнесу, тоді як кандидати можуть ставити під питання етичність компанії та її здатність забезпечити безпечні умови праці. Відновити довіру клієнтів після таких інцидентів непросто – це потребує тривалого часу та значних ресурсів на PR-кампанії та заходи з прозорості. Компанії часто змушені залучати зовнішніх експертів для перевірок безпеки та впроваджувати нові політики відкритості. У деяких випадках наслідки можуть бути настільки серйозними, що призводять до втрати ліцензії або посиленого регуляторного контролю, особливо у сфері фінансових послуг.

Особливе занепокоєння викликає те, що 85% постраждалих клієнтів діляться негативним досвідом у соціальних мережах чи серед друзів і колег. У цифрову еру такі відгуки поширюються надзвичайно швидко, завдаючи серйозного удару по репутації бізнесу в глобальному масштабі. Близько третини (33,5%) обурених клієнтів активно висловлюють своє невдоволення онлайн, що створює додатковий тиск на компанію та посилює кризову ситуацію.

Постановка задачі. Одним із пріоритетних напрямів забезпечення інформаційної стійкості є превентивне виявлення векторів атак у публічних сегментах мережі, зокрема у веб-ресурсах, серверних архітектурах та базах даних. Важливим інструментарієм у цьому контексті є методологія OSINT (Open Source Intelligence), зокрема техніка Google Dorking. Використання специфічних операторів розширеного пошуку дозволяє ідентифікувати критичні точки компрометації: незахищені адміністративні інтерфейси, конфігураційні файли з відкритим доступом та вразливі бази даних.

Попри високу аналітичну цінність, ефективна імплементація Google Dorks потребує глибокої експертизи в галузі синтаксису пошукових запитів, що обмежує їхнє масове застосування фахівцями з моніторингу безпеки. Проблема оптимізації та автоматизації процесу формування таких запитів є актуальною науково-технічною задачею, вирішення якої дозволить мінімізувати вплив людського фактора та підвищити релевантність отриманих результатів. При цьому критично важливим аспектом залишається дотримання етичних стандартів та правових норм у сфері інформаційної безпеки.

Аналіз теоретичного та практичного контексту підкреслює необхідність створення не просто інструмента для роботи з пошуковими системами, а комплексного, спеціалізованого рішення, яке дозволить ефективно шукати, виявляти та аналізувати витоки або вразливості інформації. Зростаюча кількість інцидентів, пов'язаних із випадковим чи зловмисним оприлюдненням конфіденційних даних у мережі, підкреслює актуальність такого підходу. Значна частина користувачів, особливо тих, хто має обмежений технічний досвід, не володіють зручними інструментами для оперативної перевірки доступності інформаційних ресурсів. Це обґрунтовує потребу в платформі, яка б дозволила досвідченим фахівцям і новачкам швидко та якісно визначати проблеми відкритого доступу до даних.

Основною метою роботи поставлена розробка архітектурно і функціонально складеної веб-системи, яка автоматично формує запити Google Dorks на основі заданих модулів і тегів, виконує ці запити через Google API, обробляє отримані результати, а також класифікує їх за рівнем критичності виявленої інформації.

Виклад основного матеріалу. Восени 2024 року команда Sysdig Threat Research Team виявила глобальну операцію "Emeraldwhale" [7], яка була спрямована на викриті конфігураційна файли Git, у результаті чого було викрадено понад п'ятнадцять тисяч облікових даних хмарних сервісів [8]. Під час кампанії хакери використовували приватні інструменти для зловживання неправильно налаштованими вебсервісами, саме це дозволило їм викрадати облікові дані, клонувати приватні репозиторії та вилучати хмарні облікові дані з відкритого коду. У ході операції вони зібрали більше десяти тисяч приватних репозиторіїв. Основна мета викрадення – фішинг та спам. Подальші логи зловмисників, які виявили дослідники, показали масштабну кампанію сканування спрямовану на сервери з відкритими конфігураційними файлами Git. Серед результатів атак – понад п'ятнадцять тисяч облікових даних для хмарних сервісів, більше за шістдесят тисяч URL-адрес із відкритими файлами ".git"/"config", та саме цікаве те, що близько половини із шести тисяч виявлених токенів були дійсними. Це дослідження показує, на скільки серйозними можуть бути наслідки простих помилок конфігурацій.

Стає зрозумілим, що експерти з безпеки та особливо зловмисники стараються використовувати автоматизовані сканери такого типу. Якщо потрібно швидко отримати релевантні результати на свій запит, то ручний метод пошуку буде не ефективним. Особистий досвід показав, що Google виставляє користувачеві "капчу" вже на сьомий раз відправки запиту. В масштабах кампаній сканування цього дуже мало. Підтримка Google повідомляє, що це може відбуватись через виявлення автоматичного трафіку, тобто все що торкається комп'ютерних програм, роботів, автоматизованих сервісів і засобів скрейпінгу в пошукових системах буде блокуватись. Тому постає логічне питання "як обійти зауваження від Google і не бути заблокованим?". В цьому випадку гарним інструментом буде Google Custom Search – це засіб, API якого надає Google для вільного пошуку без блокування. Таке рішення надає змогу вбудовувати функцію пошуку напряму в додаток, зберігаючи контроль над запитами та результатами без залучення браузера або взаємодії з CAPTCHA.

На рисунку 1 приведено алгоритм розгортання ситуації, коли запити до Google надходять через API (випадок з безкоштовним тарифом, без підписки).

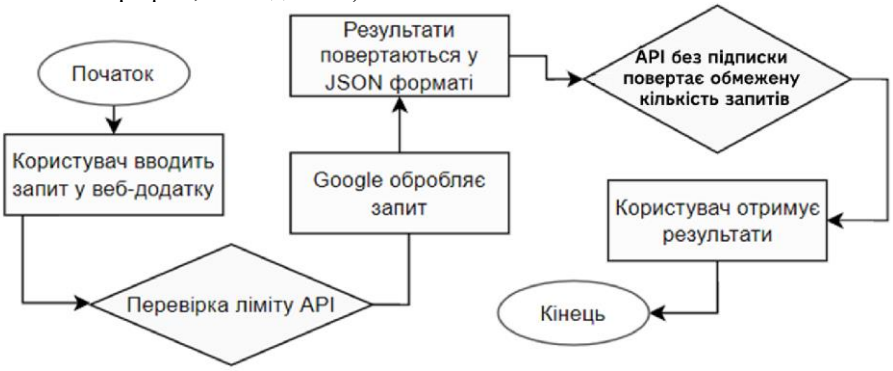


Рисунок 1. Алгоритм пошуку через Google API

Через цей факт з'являється задача організувати такий функціонал, щоб будь який користувач зміг користуватись веб-додатком без можливості отримати блокування зі сторони Google. Потрібно змоделювати ситуацію, щоб краще розуміти ризики кінцевого користувача спіймати капчу або бан: користувач отримує капчу в середньому після 6-7 запитів з інтервалом між ними в 10 секунд; нормальною поведінкою вважається 1-2 запити на хвилину; це все заважає концепції автоматизації у нашому проєкті, тому такий підхід буде виключений. Було прийнято рішення використовувати Google Custom Search API у веб-додатку цього проєкту через наступні причини:

– використання API повністю відповідає політиці Google, а це гарантує стабільність процесу та відсутність блокувань;

– навіть безкоштовний тариф дозволяє виконувати до 100 запитів на день, а платний тариф розширює можливість до 1000 і більше запитів;

– ми можемо налаштувати обмеження кількості запитів на день залежно від потреб проєкту, що забезпечує економічну ефективність;

– API повертає дані у зручному форматі (JSON або XML), це значно спрощує обробку та інтеграцію відповіді у вивід результатів.

Такий підхід відкриває можливість для обробки відповідей через машинне навчання або інші евристичні алгоритми для подальшої класифікації результатів за рівнем важливості чи загрози.

З точки зору користувача при роботі з Google dorks запитами не все може бути відразу зрозуміло, тому потрібно впровадити запити у виді шаблонів, які вже є в відкритому доступі або їх можна створити самотужки в процесі створення веб-додатку – такий підхід із шаблонами може навіть послужити своєрідним тренажером, для людей котрі мало що розуміють в техніці Google hacking; а для людей, котрі не від сьогодні знайомі із цією темою буде просто широкий вибір шаблонів, які можна легко редагувати або модифікувати в залежності від своїх цілей. А це в свою чергу економить час.

Щоб ще більше полегшити сприйняття процесу для користувача, скористаємось принципом "розділяй і володарюй". Є багато різних запитів, які ми використаємо як шаблони але всі вони розділені на дуже абстрактні категорії. Для цього проекту ж постає задача ще більше категоріїзувати шаблони, щоб кінцевий користувач міг з легкістю обрати собі найбільш потрібний. Створимо дев'ять модулів за такими задачами:

– пошук адмін-панелей;
– пошук електронних пошт;
– пошук файлів та документів;
– мета-дані різного типу;
– мережеві пристрої;
– пошук особистостей в Інтернеті;
– потенційні вразливості;
– веб-сторінки;
– веб-сервіси.

Таке розбиття на модулі забезпечує уникнення дубльованих запитів, структурує логіку пошуку та спрощує підтримку системи у майбутньому.

У підсумку на основі проведеного дослідження розроблено концепцію та архітектуру автоматизованої вебсистеми, яка за допомогою методології OSINT та інтеграції з Google Custom Search API дозволяє ефективно виявляти критичні вразливості та витоки конфіденційних даних. Головним результатом роботи є розв'язання проблеми блокувань пошукових систем та складності синтаксису запитів шляхом впровадження дев'яти спеціалізованих модулів-шаблонів, що автоматизують процес Google Dorking та класифікують результати за рівнем загрози. Це забезпечує превентивний захист інформаційних ресурсів, мінімізує вплив людського фактора та робить професійний інструментарій кібербезпеки доступним як для експертів, так і для користувачів із обмеженим технічним досвідом

Список літератури

1. Cost of a Data breach report. IBM Security. URL: https://d110erj175o600.cloudfront.net/wp-content/uploads/2023/07/25111651/Cost-of-a-Data-Breach-Report-2023.pdf (дата звернення: 09.03.2026).

2. Cost of a Data breach report 2022. Krontech. URL: https://krontech.com/highlights-of-the-2022-cost-of-a-data-breach-report (дата звернення: 09.03.2026).

3. Загальний регламент про захист даних. GDPR. URL: https://www.gdpr.org.ua/ (дата звернення: 09.03.2026).

4. Каліфорнійський закон про захист прав споживачів. eSputnik. URL: https://esputnik.com/blog/california-consumer-privacy-act (дата звернення: 10.03.2026).

5. Штраф Фейсбуку зі сторони EDPB. Edpb. URL: https://www.edpb.europa.eu/news/news/2023/12-billion-euro-fine-facebook-result-edpb-binding-decision_en (дата звернення: 10.03.2026).

6. Yahoo agree to lowered $4.48 billion deal following cyber attacks. Reuters. URL: https://www.reuters.com/article/business/verizon-yahoo-agree-to-lowered-448-billion-deal-following-cyber-attacks-idUSKBN1601EK/ (дата звернення: 10.03.2026).

7. EmeraldWhale: глобальна операція крадіжки хмарних облікових даних з конфігурацій git. Sysdig Blog. URL: https://sysdig.com/blog/emeraldwhale/ (дата звернення: 11.03.2026).

8. Витік креденшалів через GIT конфігурацію. CyberScoop. URL: https://cyberscoop.com/sysdig-git-credentials-cloud-service-emeraldwhale/ (дата звернення: 11.03.2026).

УДК 004.056

О.Т. Шаммієва, *студентка 3-го курсу*
Н.М. Якименко, *науковий керівник*
shammievalena@gmail.com, yakymenkonm@kntu.kr.ua
Центральноукраїнський національний технічний університет, Кропивницький

КОМПЛЕКСНІ ПІДХОДИ ДО ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ В СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Вступ. Стрімкий розвиток цифрової економіки, інтеграція хмарних сервісів та експоненційне зростання обсягів Big Data формують критичні виклики для сучасних систем інформаційної безпеки. Персональні дані стали одним із найцінніших активів, що робить їх пріоритетною мішенню для кіберзлочинців. Традиційні периметрові моделі захисту (інженерні загородження, інфрачервоні бар'єри, вібраційні датчики, вібраційні кабелі, радіохвильові системи) втрачають свою ефективність, що зумовлює необхідність переходу до дата-центричних підходів та архітектури нульової довіри (Zero Trust).

Технічні загрози та протидія. Однією з найбільш гострих проблем є ризик деанонімізації користувачів методами OSINT. Використовуючи аналіз метаданих (EXIF-теги у фотографіях), перехресний пошук за витоками баз даних та аналіз цифрових відбитків (Browser Fingerprinting), зловмисники здатні відновити повний профіль особи. Для протидії цим загрозам необхідно впроваджувати автоматичне очищення метаданих на етапі завантаження контенту (наприклад, за допомогою інструментів ExifTool) та засоби рандомізації параметрів браузера (антидетект-браузери, розширення, які підміняють цифрові відбитки, часовий пояс та роздільну здатність екрана, щоб уникнути ідентифікації сайтами).

Архітектурні аспекти та GDPR. З огляду на євроінтеграційні процеси, адаптація систем до вимог GDPR є технічним викликом для розробників. Це вимагає впровадження концепції Privacy by Design (такого підходу до розробки систем, продуктів та послуг, при якому захист персональних даних вбудовується безпосередньо в архітектуру з самого початку, а не додається наприкінці). Ця концепція передбачає проактивне запобігання ризикам, мінімізацію збору даних, прозорість та забезпечення максимальної приватності за замовчуванням (Privacy by Default) ще на етапі проектування. Важливим елементом є реалізація «права на забуття» через механізм криптографічного стирання (Crypto-shredding). Замість фізичного видалення даних з усіх резервних копій, знищується унікальний ключ шифрування конкретного користувача, що робить інформацію нечитабельною.

Захист біометрії та Big Data. При обробці великих масивів даних критично важливим є використання методів псевдонімізації, таких як токенізація та форматозберігаюче шифрування (FPE). Окрему увагу слід приділити біометричним даним у системах контролю доступу (СКУД). Оскільки біометрію неможливо замінити у разі компрометації, зберігання сирих зразків є неприпустимим; системи повинні оперувати лише математичними шаблонами - векторами ознак. Для захисту від спуфінг-атак (використання Deepfake чи масок) обов'язковою є імплементація технологій Liveness Detection, які визначають, чи є людина на відео живою.

Висновок. Забезпечення надійного захисту персональних даних неможливе лише організаційними заходами. Воно вимагає комплексної інтеграції стійкої криптографії (AES-256-GCM, TLS 1.3, постквантова криптографія, гібридні системи з поєднання класичних і квантово-стійких алгоритмів), безпечної обробки біометрії та адаптації інфраструктури до міжнародних стандартів, що є запорукою стабільного функціонування цифрової держави.

Список літератури

1. Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI.
2. General Data Protection Regulation (GDPR) – Regulation (EU) 2016/679.
3. NIST Special Publication 800-122. Guide to Protecting the Confidentiality of PII.
4. Stallings, W. Cryptography and Network Security: Principles and Practice. — Pearson, 2020.
5. ISO/IEC 27701:2019. Extension to ISO/IEC 27001 for privacy information management.

УДК 347.78:004.056.53(043.2)

В. О. Кукса, *бакалавр 3 курс*
Науковий керівник: О.О. Кривокульська, *ст. викладач*
7952484@stud.kai.edu.ua. olha.kryvokulska@npp.kai.edu.ua
Державний університет «Київський авіаційний інститут», Київ

КІБЕРБЕЗПЕКОВІ МЕХАНІЗМИ ЗАХИСТУ АВТОРСЬКИХ ПРАВ У ЦИФРОВОМУ СЕРЕДОВИЩІ ТА ЗАПОБІГАННЯ ІНТЕРНЕТ-ПЛАГІАТУ

Розвиток глобальної мережі інтернет докорінно змінив парадигму створення та споживання інтелектуального продукту. Сьогодні будь-який користувач може стати автором, а результати його праці миттєво стають доступними для багатомільйонної аудиторії. Проте така відкритість створює сприятливе підґрунтя для цифрового піратства та маніпуляцій з контентом.Обсяг інформації, що створюється, поширюється та зберігається через мережу інтернет стрімко зростає у сучасному цифровому середовищі. Це сприяє швидкому та доступному обміну знаннями, однак це може призводи до можливих порушень авторських прав. Переважаючою проблемою є інтернет-плагіат, який створює загрозу використання чужих праць без зазначення автора [1]. Плагіат у цифровому просторі набуває нових форм: від простого копіювання тексту (copy-paste) до складних методів перефразування за допомогою штучного інтелекту, що значно ускладнює процес ідентифікації першоджерела.

Механізми цифрової безпеки є основним рушієм захисту інтелектуальної власності в мережі. Оскільки вони займаються виявленням плагіату, запобігаючи присвоєнню інтелектуальних доробків. Ефективна система захисту повинна бути багаторівневою, поєднуючи превентивні заходи, методи активного моніторингу та юридично значущі докази авторства.

Таблиця1.
Проблема плагіату

Види плагіату	Характеристика	Технологічне рішення	Результат застосування
Інтернет-плагіат	Використання чужих матеріалів без посилання на першоджерело	Перевірка унікальності текстів	Виявлення фрагментів, що не є автентичними
Порушення авторських прав	Поширення або копіювання файлу без зазначення автора	Цифрові водяні знаки	Ідентифікація першоджерела
Втрата інформації про автора	Без закріплення інформації за автором першоджерело може загубитися у мережі	Блокчейн-технології	Незмінна інформація про дату та час створення
Недостатня обізнаність користувачів	Користувачі можуть не дотримуватися правил академічної доброчесності	Освітні та правові норми	Мінімізація плагіату

Завданням дослідження є аналіз основних кібербезпекових механізмів, що використовуються для захисту авторських прав у цифровому середовищі та запобігання інтернет-плагіату. Актуальність роботи підкріплюється необхідністю захисту не лише текстового контенту, а й мультимедійних даних, програмного коду та цифрових активів.

Особлива увага акцентується на розвитку сучасних технологій, які мають забезпечувати протидію неправомірному присвоєнню інформації, а також захисту авторських прав та запобіганню плагіату в мережі. Застосування методів інтелектуального аналізу даних (Data Mining) дозволяє автоматизувати пошук запозичень навіть у глибоко модифікованих матеріалах.

За допомогою перевірки унікальності текстів здійснюється боротьба з плагіатом в мережі. Перевірка відбувається зокрема через порівняння документів з існуючими базами даних публікацій, таким чином тексти аналізуються на подібність між собою та виявляються фрагменти, що не є унікальними. Сучасні алгоритми використовують метод шинглів (shingle method) та семантичний аналіз, що дозволяє знаходити збіги не лише за послідовністю слів, а й за змістом. У закладах освіти академічна доброчесність забезпечується за допомогою подібних технологій.

Захист авторських прав також може відбуватися через використання цифрових водяних знаків. Механізм полягає у додаванні прихованої інформації про джерело на оригінальному тексті. На відміну від видимих логотипів, стеганографічні методи дозволяють вбудовувати мітку безпосередньо у структуру цифрового файлу — у найменш значущі біти зображення чи аудіосигналу. Ідентичність буде підтверджено навіть у разі

копіювання або поширення файлу мережею, оскільки це буде маркером першоджерела. Широко використовується в інтернеті також на аудіо-та відеофайлах. Таким чином саме поєднанням криптографічних методів забезпечується захист цифрових даних. Крім того, це створює надійну доказову базу при розгляді суперечок у судовому порядку.

Блокчейн застосовується у кіберзахисті цифрового контенту для закріплення справжнього автора, шляхом впровадження незмінних записів про дату та час створення публікації. Використання децентралізованого реєстру дозволяє створити «цифровий відбиток» (хеш-суму) твору. Оскільки хеш є унікальним для кожного файлу, будь-яка спроба змінити контент призведе до зміни хешу, що миттєво викриє підробку. Технологія алгоритму стає все більш перспективною у використанні для протидії плагіату. Це гарантує зберігання інформації у первісному вигляді з дотриманням авторських прав[2]. Смарт-контракти в мережі блокчейн можуть також автоматично регулювати виплату роялті авторам при кожному використанні їхнього твору.

Необхідно підвищувати рівень обізнаності користувачів мережі про дотримання авторських прав і запобігання плагіату. Кібергігієна та етика цифрової взаємодії мають стати частиною загальної медіаграмотності суспільства. Правові та освітні норми з долученням передових технологій кібербезпеки мінімізують випадки застосування плагіату, забезпечуючи більш результативний захист у цифровому середовищі. Створення єдиного міжнародного правового поля для захисту цифрових активів є ключовим викликом сучасності.

Таблиця 2.

Алгоритми забезпечення політики антиплагіату

Алгоритм	Суть технології	Де використовується	Переваги
Перевірка унікальності тексту	Порівняння текстів між собою на предмет виявлення подібностей	Заклади освіти, наукові публікації	Виявляє плагіат і забезпечує академічну доброчесність
Цифрові водяні знаки	Додавання прихованої інформації про першоджерело на автентичному тексті	Цифрові документи, зображення	Зберігає ідентичність навіть після копіювання та поширення файлу
Блокчейн	Закріплення оригінального авторства за допомогою незмінних записів про першоджерело	Платформи публікацій, цифрові архіви	Унеможливлює здійснення підробок

Висновок

Збереження ідентичності публікації та першоджерела є передовим завданням сучасної кіберзпеки. В умовах тотальної цифровізації захист інтелектуальної власності виходить за межі суто юридичної площини, стаючи об'єктом інженерних рішень. Перевірка унікальності текстів, застосування цифрових водяних знаків, блокчейн, криптографічні методи захищають авторські права з посиланням на оригінал в автентичному вигляді та унеможливлюють неправомірне використання інформації. В умовах розвитку цифрових технологій зростає необхідність розповсюдження політики антиплагіату. Лише узагальнене використання всіх механізмів кіберзахисту може створити підхід до протидії плагіату в мережі. Подальші дослідження повинні бути спрямовані на вдосконалення методів захисту динамічного контенту та розробку інструментів боротьби з плагіатом, згенерованим нейромережами.

Список літератури:

1.Stallings W. *Effective Cybersecurity: A Guide to Using Best Practices and Standards.* Addison-Wesley, 2018.
2.Kshetri N. *Blockchain and Intellectual Property Protection.* IEEE Computer, 2018.

УДК 004[89::(056.53+413.4)]

В.В. Цуркан[1,2], Є.О. Вербова[1]
v.v.tsurkan@gmail.com, evgeniaverbova25@gmail.com
[1]*Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ*
[2]*Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова Національної академії наук України, Київ*

АНАЛІЗ ЗЛОВЖИВАНЬ ІНТЕГРУВАННЯМ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ЗІ ЗОВНІШНІМИ СЕРВІСАМИ

Одним з найбільш небезпечних ризиків кібербезпеки великих мовних моделей (англ. large language model, LLM) є упроваджування запитів (англ. prompt injection) [1, 2]. Його настання може призводити до непередбачуваних змінень поведінки та результатів їх використання. Це пов'язується з уразливостями обробляння і передавання запитів великими мовними моделями. Експлуатування загрозами таких уразливостей орієнтоване на отримання шкідливого контенту, несанкціонованого доступу та впливання на результати використання. Серед загроз виокремлюються безпосереднє (пряме, англ. direct prompt injections) та опосередковане (непряме, англ. indirect prompt injection) впроваджування запитів. У першому випадку джерелом є користувач або зловмисник, а в другому – файли або прикладні застосунки. З останнім асоціюється розширюваність функційності великих мовних моделей. Вона досягається інтегрованістю зі зовнішніми сервісами [1]. На практиці така інтегрованість може спонукати до зловживань ними (англ. tool abuse), зокрема, виникання нових векторів реалізувань загроз і, як наслідок, збільшування поверхні атаки великих мовних моделей. Істинність даного твердження підтверджується відомими результатами досліджень, наприклад, скасування обмежень (англ. jailbreak), відмовляння від шкідливих запитів (англ. refusal-trained). Тож аналізування зловживань інтегруванням великих мовних моделей зі зовнішніми сервісами є актуальним завданням.

Програмні застосунки, платформи, служби типізуються як зовнішні сервіси, наприклад: поштові системи, системи вебаналітики, платіжні системи. Кожна з них структурно функціонує поза великою мовною моделлю і орієнтована на розширювання її функційності. Попри це інтегрування зі зовнішніми сервісами може призводити до їх навмисного «правильного» використання через, наприклад, лист, календар, квитанцію, документ. Воно реалізується на етапі початкового доступу (англ. initial access) ланцюга кіберзнищення (англ. cyber kill chain) [3]. Наприклад [4], LLM-агент інтегрується з корпоративною поштовою системою. Це дозволяє автоматично узагальнювати вхідні електронні листи. Наведений приклад інтегрування передбачає отримання доступу до функціоналу поштової скриньки. Йдеться про читання, видаляння або відправляння повідомлень. Зловмисник може надіслати електронного листа з вкладеним «безпечним» зображенням. Користувач після звернення до LLM-агента отримує узагальнення останніх вхідних листів. При оброблянні сформованого запиту великою мовною моделлю аналізується їхній вміст з вкладеннями. Внаслідок дослідження «безпечного» зображення можуть приховано викликатися зовнішні сервіси як варіант [4]: <function.delete_email which=«all»>. За результатами такого зловживання видаляються усі електронні листи користувача. Наведеним наочним прикладом демонструються негативні наслідки відсутності механізмів валідування вихідних даних великої мовної моделі та обмеження прав інтегрування зі зовнішніми сервісами.

Отже, попри розширювання функційних можливостей великих мовних моделей інтегрування зі зовнішніми сервісами призводить до виникання нових векторів реалізувань загроз і, як наслідок, збільшування поверхні атаки. Це досягається використанням програмних застосунків, платформ, служб для початкового доступу в межах ланцюга кіберзнищення. Зокрема навмисного «правильного» використання великих мовних моделей.

Список літератури

1. LLM01:2025 Prompt Injection. 2025 Top 10 Risk & Mitigations for LLMs and Gen AI Apps. URL: https://genai.owasp.org/llmrisk/llm01-prompt-injection/ (accessed on: 20.03.2026).

2. Вербова Є.О., Цуркан В.В. Аналіз кіберризику отруєння даних великих мовних моделей. *Інженерія програмного забезпечення і передові інформаційні технології (Soft Tech-2025)* : матеріали IX міжнародної науково-практичної конференції молодих вчених та студентів (Київ, 26–28 листопада 2025 року). Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», ФІОТ, 2025. С. 33–36.

3. Brodt O., Feldman E., Schneier B., Nassi B. The Promptware Kill Chain : How Prompt Injections Gradually Evolved Into a Multistep Malware Delivery Mechanism. *Computer Science : Cryptography and Security*. 2026. arXiv:2601.09625v2. DOI: https://doi.org/10.48550/arXiv.2601.09625.

4. Xiaohan Fu, et al. Misusing Tools in Large Language Models with Visual Adversarial Examples. *Computer Science : Cryptography and Security*. 2023. arXiv:2310.03185. DOI: https://doi.org/10.48550/arXiv. 2310.03185.

УДК 342.738+004.056.5:004.738.5

О.С. Ткаченко, *аспірант кафедри Кібербезпеки*,
А.В. Ільєнко, *канд. техн. наук, доцент, завідувач кафедри Кібербезпеки*
alexsunnik@gmail.com, anna.ilienko@npp.kai.edu.ua
Державний університет «Київський авіаційний інститут», Київ, Україна

ЛАНДШАФТ ВРАЗЛИВОСТЕЙ ТА ІНФРАСТРУКТУРА КІБЕРЗАГРОЗ В СИСТЕМАХ СОЦІАЛЬНИХ МЕРЕЖ

Вразливості в екосистемах соціальних мереж виникають на небезпечному перетині психологічних слабкостей користувачів та недоліків технічної архітектури платформ. Аналіз глобальних звітів з кібербезпеки за 2024–2025 роки розкриває безпрецедентну ескалацію складних, багатовекторних загроз.

Соціальна інженерія остаточно закріпила статус найефективнішого методу обходу традиційних технічних засобів контролю кібербезпеки. Відповідно до даних *2025 Unit 42 Global Incident Response Report*, соціальна інженерія стала вектором початкового доступу номер один, ініціювавши 36% усіх розслідуваних інцидентів. Загалом, 68% усіх порушень безпеки даних у світі безпосередньо пов'язані з людським фактором, який включає фішингові кампанії, використання вкрадених облікових даних та ненавмисні помилки персоналу. Такі атаки відрізняються катастрофічною результативністю: 60% успішних вторгнень, заснованих на соціальній інженерії, завершуються експозицією конфіденційних даних, що на 16 відсоткових пунктів вище, ніж при використанні інших векторів атак.

У 2025 році спостерігаються дві паралельні, але вкрай руйнівні тенденції. По-перше це використання генеративного ШІ (GenAI). Технології штучного інтелекту радикально знизили вартість та підвищили реалістичність атак. Зловмисники масово застосовують великі мовні моделі (LLM), натреновані на архівах публічних дописів жертви у соціальних мережах, для генерації гіперперсоналізованих фішингових повідомлень, які бездоганно імітують стиль письма, лексикон та інтереси конкретної людини. Глибокі підробки (Deepfakes) стали стандартним інструментом для проведення складних "романтичних афер" (Romance scams) та кампаній з сексторції (sextortion) через відеодзвінки у прямих повідомленнях. І по-друге це High-Touch компрометації проти автоматизованого обману. З одного боку, існують високоорганізовані угруповання (як-от фінансово мотивована група Muddled Libra), які проводять так звані "High-Touch" атаки. Вони збирають інформацію з професійних мереж типу LinkedIn для створення бездоганних легенд, а потім у реальному часі маніпулюють співробітниками IT-підтримки (help desks) для скидання параметрів багатофакторної аутентифікації (MFA).[44] З іншого боку, спостерігається лавина "At-Scale" автоматизованих обманів, які покладаються на тактики "ClickFix", спуфінг системних повідомлень браузера та отруєння пошукової видачі (SEO poisoning) для інфікування користувачів без традиційних фішингових листів.

Фінансовий мотив залишається домінуючим рушієм. Лише у США у 2024 році споживачі втратили понад 12.5 мільярдів доларів через онлайн-шахрайство, при цьому глобальна середня вартість ліквідації наслідків витоку даних для організацій варіюється від 4.4 до 4.88 мільйона доларів США за інцидент.

Незважаючи на мільярдні інвестиції в кібербезпеку, прикладні програмні інтерфейси (API) соціальних мереж залишаються основним джерелом масових, багатомільйонних витоків даних. Проблема "Тіньових API" (Shadow APIs) — застарілих, недокументованих або неправильно сконфігурованих інтерфейсів — стала однією з найгостріших загроз 2025 року. Скрейпінг (Scraping), тобто автоматизований збір масивів даних через API платформи, стирає тонку межу між "публічною доступністю" інформації та її "масовою компрометацією". Два інциденти 2024-2025 років ілюструють масштаб проблеми:

1. Криза з API Facebook (Травень 2025). На спеціалізованих хакерських ресурсах було виявлено колосальну базу даних, що містила записи 1.2 мільярда користувачів Meta (Facebook). Зловмисники успішно експлуатували вразливий публічний API платформи, щоб шляхом масового скрейпінгу агрегувати імена, ідентифікатори Facebook ID, електронні адреси, номери телефонів, геолокаційні дані та дати народження. Незважаючи на заяви корпорації про те, що дані стосуються історичних витоків, сам факт можливості збору такої кількості інформації демонструє критичні ризики надмірного надання привілеїв кінцевим точкам API (over-permissive endpoints) та відсутності адекватного лімітування.

2. Витік бази даних Instagram (Січень 2026). На сумнозвісному даркнет-майданчику BreachForums у безкоштовному доступі була опублікована база з 17.5 мільйонами записів користувачів Instagram у структурованих форматах JSON/TXT. З них понад 6.2 мільйона записів містили актуальні електронні адреси та часткові геолокаційні координати. Цей інцидент стався не внаслідок традиційного проникнення на сервери Meta, а через експлуатацію архітектури API, яка дозволяла агрегувати розрізнені фрагменти публічної та напівпублічної інформації у єдину, ідеально підготовлену для фішингу базу. Наслідками стали глобальні хвилі автоматизованих спроб брутфорс-доступу та скидання паролів мільйонів користувачів.

Зловмисники успішно обходять захисні бар'єри соціальних платформ (такі як обмеження швидкості запитів — Rate Limiting) за допомогою масивних інфраструктур проксі-серверів. Вони розподіляють

автоматизовані запити між мільйонами реальних резидентних IP-адрес (які належать звичайним користувачам інтернету), роблячи традиційне блокування на основі IP абсолютно неефективним — явище, яке інженери з кібербезпеки називають "смертю від тисячі легітимних запитів" (Death by a Thousand Legitimate Requests).

Соціальні мережі активно експлуатуються кіберзлочинцями як вектори для розгортання ботнетів та дистрибуції шкідливого програмного забезпечення (ШПЗ). Згідно з дослідженнями, проведеними на початку 2025 року, безпрецедентні 60% інцидентів з розповсюдженням шкідливого ПЗ були прямо пов'язані з цифровою рекламою (Malvertising), яка транслювалася на платформах соціальних медіа та новинних сайтах. Автоматизовані системи дистрибуції реклами використовуються для ін'єкції шкідливого коду, який інфікує пристрій користувача навіть без необхідності натискати на банер (техніка drive-by downloads).

У контексті парадигми Соціального Інтернету речей (Social Internet of Things, SIoT), де пристрої (смарт-годинники, камери, роутери) здатні самостійно формувати "соціальні" зв'язки для обміну даними, поширення ШПЗ моделюється за допомогою систем звичайних диференціальних рівнянь (ODEs), що враховують вплив хабів та фізичних хот-спотів (hotspots). Емпіричні дослідження доводять, що щільність соціальних зв'язків між пристроями експоненціально прискорює зараження. Симуляції показують, що збільшення кількості динамічних відносин між об'єктами з двох до чотирьох скорочує час повного інфікування всієї мережі на 45%, що створює критичні загрози для концепції розумних міст.

Для протидії ботнетам (таким як сумнозвісний Mirai та його більш досконалі мутації Satori і Masuta), які здатні генерувати терабітні DDoS-атаки та викрадати криптовалюти, сучасні системи виявлення еволюціонували від сигнатурного аналізу пакетів до аналізу графової структури мережевого трафіку. Для цих цілей широко застосовуються графові нейронні мережі (Graph Neural Networks, GNNs).

Проте, традиційні GNN мають фундаментальний недолік: вони надмірно залежать від топологічної структури мережі, на якій проходили навчання, що призводить до проблеми "перенавчання" (overfitting). При спробі виявити ботнет у невідомій мережі з іншою топологією, їхня ефективність стрімко падає. Для вирішення цієї інженерної проблеми була розроблена інноваційна методологія SIR-GN (Structural Iterative Representation for Graph Nodes). Ця інференційна модель фокусується виключно на внутрішніх структурних властивостях вузла, обчислюючи його репрезентацію через серію ітерацій:

1. Представлення вузла ініціалізується та нормалізується.

2. Проводиться кластеризація за алгоритмом K-Means на основі заданого гіперпараметра розмірності кластера (n_c).

3. Обчислюється ймовірність приналежності вузла до кожного конкретного кластера на основі геометричної відстані до центроїда.

4. Значення структурної агрегації оновлюється з урахуванням багатокрокових (multi-hop) взаємозв'язків сусідів, створюючи унікальний структурний вектор для кожного пристрою.

Експериментальне оцінювання цієї моделі продемонструвало вражаючі результати. У поєднанні з класифікаторами типу Random Forest, SIR-GN досягає феноменальної здатності до узагальнення. Під час тестування на реалістичних даних атак через P2P-мережі (Peer-to-Peer), алгоритм SIR-GN, натренований на зовсім іншій топології (наприклад, Chord або Leet), продовжував ідентифікувати ботів з оцінкою F1 на рівні 97.0–98.0%. Водночас альтернативні сучасні методи, такі як ABD-GN, демонстрували катастрофічне падіння ефективності, набираючи від 0.0% до 2.5% точності на незнайомих топологіях графа.

Захисні механізми та криптографічні методи збереження конфіденційності

Відповіддю на стрімке зростання масштабів витоків даних та вразливостей є еволюція захисних механізмів, які намагаються знайти компроміс між доступністю послуг, якістю роботи алгоритмів рекомендацій та збереженням приватності користувачів.

Обхід обмежень та анти-скрейпінг системи

Сучасні комерційні скрейпери використовують надзвичайно витончені техніки обходу систем захисту ОСМ у 2025 році:

- Використання інфраструктури, що маршрутизує запити через пули з понад 90 мільйонів реальних IP-адрес користувачів домашнього інтернету. При отриманні помилки HTTP 429 система автоматично та миттєво переключає запит на нову "чисту" IP-адресу.

- Рандомізація та маніпуляція заголовками. Зловмисники динамічно змінюють такі HTTP-заголовки, як User-Agent, Accept-Language, а також маніпулюють полями X-Forwarded-For та X-Forwarded-Host, щоб зімітувати абсолютно унікальні клієнтські сесії та обдурити системи лімітування.

- Використання спеціалізованих хмарних браузерів на базі модифікованого рушія Chromium. Ці браузери мають вбудовані антидетект-механізми, які симулюють реалістичні рухи миші, рендерять важкий динамічний JavaScript і успішно вирішують криптографічні анти-бот перевірки (CAPTCHA) без участі людини.

Для протидії цьому сучасні захисні системи та Web Application Firewalls відмовилися від блокування за IP на користь механізмів Device Fingerprinting (DFP). Цей метод збирає сотні атрибутів клієнта (роздільна здатність екрана, унікальні шрифти, особливості Canvas рендерингу, параметри ОС) для створення незмінного криптографічного відбитка пристрою, який дозволяє ідентифікувати бота незалежно від того, яку проксі-адресу він використовує в даний момент.

Криза довіри до платформ соціальних мереж, спричинена скандалами на кшталт Cambridge Analytica, де приватні дані понад 87 мільйонів користувачів були використані для психографічного таргетування, стимулювала наукову спільноту до впровадження криптографічних технологій аналізу.

Федеративне навчання (Federated Learning, FL)

Це парадигма децентралізованого машинного навчання. Замість того, щоб передавати сирі особисті дані (повідомлення, фотографії, геодані) на центральні сервери соціальної мережі для навчання рекомендаційних алгоритмів, моделі штучного інтелекту тренуються безпосередньо на смартфонах кінцевих користувачів. Кожен пристрій обчислює лише локальні оновлення ваг моделі і відправляє їх на сервер, де вони агрегуються (наприклад, за алгоритмом FederatedAveraging) для створення покращеної глобальної моделі. Це фундаментально знижує ризик перехоплення приватних даних in-transit.

Диференційна приватність (Differential Privacy, DP)

Хоча Федеративне навчання захищає самі дані, дослідники довели, що звичайних оновлень ваг моделі достатньо для здійснення атак зворотного реверс-інжинірингу та відтворення оригінальних повідомлень користувача. Для усунення цієї вразливості застосовується Диференційна приватність. Цей математичний фреймворк гарантує безпеку шляхом навмисного введення каліброваного статистичного шуму (через розподіли Лапласа, Гаусса або експоненціальні механізми) до градієнтів перед їхньою відправкою на центральний сервер. Математично система забезпечує рівень захисту, визначений параметром ε (privacy budget): алгоритм розроблений так, щоб присутність або відсутність даних будь-якого одного конкретного користувача в загальному датасеті не змінювала результат аналітичного запиту з ймовірністю, що перевищує фактор e^{ε}. Основний технологічний виклик полягає у пошуку ідеального балансу між рівнем шуму (для гарантії приватності) та збереженням функціональної корисності (accuracy) аналітичної моделі.

Стійка анонімізація графів

Для бізнес-аналітики та алгоритмів кластеризації у соціальних мережах широко використовуються гібридні підходи (такі як методики k-анонімності та l-різноманітності, адаптовані до графових структур). Ці алгоритми динамічно розщеплюють вузли з чутливими персональними даними на кілька псевдовузлів, навмисно модифікуючи локальну структуру графа. Це унеможливлює зворотну ідентифікацію конкретної особи за її зв'язками чи інтересами, зберігаючи при цьому загальну макро-топологію мережі для потреб глобальної аналітики.

Таким чином, сучасний ландшафт онлайн-соціальних мереж являє собою одну з найскладніших категорій розподілених інформаційних систем. Структурна сила цих мереж, яка полягає у відкритості та здатності до безперешкодної передачі інформації через систему хабів та слабких зв'язків, одночасно формує їхню головну макроструктурну вразливість. Зловмисні актори, що використовують автоматизацію, генеративний штучний інтелект та експлуатують психологічні механізми (такі як соціальний доказ та синдром втрачених можливостей), здатні інфікувати соціальні графи та порушувати інформаційну безпеку з суб-лінійною ефективністю. У контексті ескалації кіберзагроз, що охоплюють соціальну інженерію, масовий скрейпінг API та поширення ботнетів, майбутнє стійкості (resilience) цих соціотехнічних систем неминуче залежатиме від глобального переходу до парадигм безпеки "за задумом" (security-by-design), включно з імплементацією структурного аналізу графів (на кшталт SIR-GN), децентралізованого федеративного навчання та строгих математичних гарантій диференційної приватності.

Список літератури

1. 2025 Unit 42 Global Incident Response Report: Social Engineering Edition. Palo Alto Networks. URL: https://unit42.paloaltonetworks.com/2025-unit-42-global-incident-response-report-social-engineering-edition/ (дата звернення: 18.03.2026).

2. 100+ Latest Social Engineering Statistics: Costs, Trends, AI [2025]. Sprinto. URL: https://sprinto.com/blog/social-engineering-statistics/ (дата звернення: 18.03.2026).

3. 139 Cybersecurity Statistics and Trends [updated 2025]. Varonis. URL: https://www.varonis.com/blog/cybersecurity-statistics (дата звернення: 18.03.2026).

4. The Easy Way In/Out: Securing The Artificial Future, Trend Micro Security Predictions for 2025. Trend Micro. URL: https://www.trendmicro.com/vinfo/us/security/research-and-analysis/predictions/the-artificial-future-trend-micro-security-predictions-for-2025 (дата звернення: 18.03.2026).

5. Gen Q4/2025 Threat Report. Gen Digital. URL: https://www.gendigital.com/blog/insights/reports/threat-report-q4-2025 (дата звернення: 18.03.2026).

6. The Latest Cyber Crime Statistics (updated October 2025). AAG IT Support. URL: https://aag-it.com/the-latest-cyber-crime-statistics/ (дата звернення: 18.03.2026).

7. How Hackers Exploit Shadow APIs (Case Studies from 2025 Breaches). IntelligenceX. URL: https://blog.intelligencex.org/how-hackers-exploit-shadow-apis-case-studies-from-2025-breaches (дата звернення: 18.03.2026).

8. Global Data Breaches and Cyber Attacks in May 2025 – More Than 1.4 Billion Records Breached. GRC Solutions. URL: https://grcsolutions.io/global-data-breaches-and-cyber-attacks-in-may-2025-more-than-1-4-billion-records-breached/ (дата звернення: 18.03.2026).

9. Gupta D. Instagram Data Leak 2026: 17.5M Users & API Security Failures. Deepak Gupta. URL: https://guptadeepak.com/the-instagram-api-scraping-crisis-when-public-data-becomes-a-17-5-million-user-breach/ (дата звернення: 18.03.2026).

10. API Rate Limiting Fails: Death by a Thousand (Legitimate) Requests / InstaTunnel. Medium. URL: https://medium.com/@instatunnel/api-rate-limiting-fails-death-by-a-thousand-legitimate-requests-30e24aba8b7f (дата звернення: 18.03.2026).

11. Online Ads Account for 60% of Malware Spread in 2025. Cybersecurity Insiders. URL: https://www.cybersecurity-insiders.com/online-ads-account-for-60-of-malware-spread-in-2025/ (дата звернення: 18.03.2026).

12. Miura H., Kimura T., Hirata K. Modeling of Malware Propagation in Wireless Mobile Networks with Hotspots Considering the Movement of Mobile Clients Based on Cosine Similarity. *Electronics*. 2025. Vol. 14, No. 17. P. 3528. DOI: https://doi.org/10.3390/electronics14173528.

13. Al Kindi A., Al Abri D., Al Maashri A., Bait-Shiginah F. Analysis of malware propagation behavior in Social Internet of Things. *International Journal of Communication Systems*. 2019. Vol. 32, No. 15. P. e4102. DOI: https://doi.org/10.1002/dac.4102.

14. Cuzzocrea A., Hafsaoui A., Gallo C. Detecting and Analyzing Botnet Nodes via Advanced Graph Representation Learning Tools. *Algorithms*. 2025. Vol. 18, No. 5. P. 253. DOI: https://doi.org/10.3390/a18050253.

15. Bypass Rate Limit While Web Scraping Like a Pro. Scrapeless. URL: https://www.scrapeless.com/en/blog/web-scraping-rate-limit (дата звернення: 18.03.2026).

16. Top strategies to prevent web scraping and protect your data. Stytch. URL: https://stytch.com/blog/web-scraping/ (дата звернення: 18.03.2026).

17. Biggest Data Breaches in US History (Updated 2025). UpGuard. URL: https://www.upguard.com/blog/biggest-data-breaches-us (дата звернення: 18.03.2026).

18. Shalabi E., Khedr W., Rushdy E., Salah A. A Comparative Study of Privacy-Preserving Techniques in Federated Learning: A Performance and Security Analysis. *Information*. 2025. Vol. 16, No. 3. P. 244. DOI: https://doi.org/10.3390/info16030244.

19. Jahani K., Moshiri B., Hossein Khalaj B. PPFL: Privacy-Preserving Techniques in Federated Learning. *Journal of Artificial Intelligence, Applications and Innovations*. 2024. Vol. 1, No. 3. P. 49–67. DOI: https://doi.org/10.61838/jaiai.1.3.6.

20. Alabi M., Ovais A. Federated Learning with Differential Privacy: Preserving Data Privacy in Collaborative Learning. 2024. URL: https://www.researchgate.net/publication/383254578_Federated_Learning_with_Differential_Privacy_Preserving_Data_Privacy_in_Collaborative_Learning (дата звернення: 18.03.2026).

21. More P., Tiwari V. A Survey on Privacy Preservation Techniques in Social Clustering via Federated Learning and Deep Learning. *International Journal of Computational Intelligence and Applications*. 2026. DOI: https://doi.org/10.1142/S1469026826300028.

УДК 004.45:004.056

О.К. Конопліцька-Слободенюк , В.В. Савельєв, А.С. Коваленко
ksuha80@gamail.com vovasavelev099@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

КОМПЛЕКСНІ МЕТОДИ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ: ВИКЛИКИ ШТУЧНОГО ІНТЕЛЕКТУ ТА SIEM-ТЕХНОЛОГІЇ

Сьогодні ми бачимо, як інформація про нас стала найдорожчим товаром у світі. Кожен клік, кожна покупка в інтернеті чи візит до лікаря залишають величезний цифровий слід. Хакери давно не ламають сайти просто заради забави — вони цілеспрямовано полюють на бази даних, щоб потім їх продати або використати для шантажу. Через війну в Україні ця проблема взагалі вийшла за межі звичайної приватності та стала критичним питанням національної безпеки. Якщо якась база даних потрапить до рук ворога, це може коштувати комусь життя. Саме тому держава зараз дозволяє ховати критично важливі реєстри в іноземних хмарних сервісах, де рівень фізичного та програмного захисту значно вищий.

А тим часом звичайні шахраї продовжують активно штампувати фейкові сайти. Вони маніпулюють темами державних виплат чи допомоги від благодійних фондів. Їхня мета банальна — змусити довірливих людей самостійно віддати дані своїх банківських карток та паролі. Ще один величезний головний біль сьогодення — це специфіка роботи медіа. Дуже часто журналісти чи блогери публікують детальну інформацію про військовополонених, жителів окупованих територій або навіть дітей, не надто замислюючись про наслідки. Знайти ту саму золоту середину між суспільним інтересом, свободою преси та правом конкретної людини на абсолютну приватність буває вкрай важко.

ТРАНСФОРМАЦІЯ НОРМАТИВНО-ПРАВОВОГО ЗАБЕЗПЕЧЕННЯ

Якщо уважно подивитися на сучасні закони, то вимоги до бізнесу стають дедалі жорсткішими по всьому світу. Звісно, головним прикладом та орієнтиром для нас залишається європейський регламент GDPR. Компанії в ЄС вже отримують багатомільйонні штрафи за витоки інформації, і це чудово стимулює бізнес інвестувати в безпеку. Але подібні процеси йдуть скрізь. От, наприклад, в Австралії нещодавно оновили законодавство і змусили компанії обов'язково пояснювати клієнтам, якщо якесь рішення щодо них (наприклад, видача кредиту) ухвалює не людина, а комп'ютерний алгоритм. У США досі немає одного загального закону на всю країну, кожен штат придумує щось своє, і міжнародним компаніям від цього дуже непереливки. Україна теж намагається швидко підтягуватися під ці цивілізовані стандарти.

АРХІТЕКТУРА БЕЗПЕКИ: PRIVACY BY DESIGN ТА ОЦІНКА РИЗИКІВ

Найцікавіше те, що самі по собі закони нікого не захистять від зламу. Ці юридичні вимоги треба вбудовувати безпосередньо в код програм. Саме це і є концепцією «Privacy by Design» (приватність за дизайном). Ідея дуже проста: програміст повинен думати про шифрування та безпеку ще тоді, коли тільки малює архітектуру майбутньої системи, а не тоді, коли продукт вже пішов у реліз. Більше того, будь-яка програма чи сайт має працювати так, щоб за замовчуванням збирати лише той абсолютний мінімум інформації, без якого вона просто не зможе функціонувати. Цей принцип називають «Privacy by Default».

Щоб ці підходи реально працювали на практиці, європейські стандарти вимагають обов'язкового проведення процедури DPIA (Оцінка впливу на захист даних). Якщо компанія хоче запустити щось потенційно небезпечне, наприклад, складний алгоритм для масштабного профілювання покупців чи аналізу їхньої поведінки, вона зобов'язана спочатку зробити таку перевірку. Це дозволяє знайти архітектурні проблеми та можливі шляхи витоку інформації ще до того, як додатком почнуть масово користуватися.

ЧИМ ЗАХИЩАТИСЯ НА ПРАКТИЦІ: SIEM-СИСТЕМИ ТА АНАЛІТИКА

Звичайні антивіруси чи файрволи вже давно не працюють так, як треба. Для того, щоб реально контролювати сервери та виявляти загрози, середній та великий бізнес зараз масово розгортає комплексні SIEM-системи. Практика показує, що платформи на зразок Splunk, IBM QRadar, Elastic Security або Microsoft Sentinel стали абсолютно необхідними для підтримки безпеки. Такі платформи давно перестали бути просто збирачами текстових логів. Вони бачать усю корпоративну мережу цілком: від хмарних серверів до робочих ноутбуків співробітників, які працюють з дому.

Найкорисніша функція сучасних SIEM — це поведінкова аналітика на базі машинного навчання (UEBA). Система тижнями мовчки запам'ятовує, як зазвичай працює конкретний співробітник. Якщо звичайний менеджер раптом посеред ночі вирішить завантажити всю базу клієнтів на свій особистий диск, алгоритм миттєво зрозуміє, що це підозріла аномалія, і просто заблокує йому доступ.7 На сьогодні це чи не єдиний дієвий спосіб боротися з внутрішніми витоками інформації та зламаними акаунтами.

ВПЛИВ ШТУЧНОГО ІНТЕЛЕКТУ: ВІД ПОМІЧНИКА ДО ГОЛОВНОЇ ЗАГРОЗИ

Найбільший виклик останніх років — це, безумовно, стрімкий розвиток штучного інтелекту. З одного боку, це надзвичайно зручний інструмент. Дизайнери та розробники виконують рутинну роботу в рази швидше, прототипуючи інтерфейси в Uizard або генеруючи тексти через ChatGPT. Але тут криється величезна загроза безпеці. Співробітники часто необачно завантажують у ці публічні нейромережі конфіденційну інформацію своїх клієнтів або закритий програмний код. Ці чутливі дані миттєво опиняються на чужих серверах, і ніхто не знає, як вони там будуть використовуватися для навчання наступних моделей.

Ситуація стає ще гіршою, якщо подивитися на автономних ШІ-агентів (Agentic AI). Вони вже вміють самостійно шукати інформацію в інтернеті, відправляти листи і навіть працювати з платіжними системами.9 Якщо раніше хакери писали складний код, щоб зламати систему, то зараз їм достатньо написати правильне речення (промпт) англійською мовою. Вони просто просять ШІ-агента проігнорувати правила безпеки та видати базу паролів. Якщо хакер зможе так обманути агента, він отримає доступ до всього. Експерти вже визначили це як одну з найбільших загроз майбутнього, класифікуючи такі атаки як ін'єкції промптів та появу «зловмисних» агентів.

Щоб цьому запобігти, науковці створюють спеціальні програми-контролери. Наприклад, з'явилася система AudAgent, яка в режимі реального часу постійно стежить за тим, щоб ШІ дотримувався політики приватності під час виконання своїх завдань. Корпорація Microsoft також випустила спеціальний набір інструментів (Agent Governance Toolkit) для жорсткого обмеження прав таких автономних агентів. Втім, штучний інтелект можна використовувати і для захисту. Наприклад, передові моделі (як-от Claude Mythos) зараз активно використовуються провідними корпораціями для пошуку вразливостей нульового дня у програмному коді, і роблять вони це набагато швидше та якісніше за людей-тестувальників.

Ще одне дуже слабке місце сучасних офісів — це корпоративна телефонія. Звичайних дротів майже не залишилося, всі компанії дзвонять через інтернет за технологією VoIP.14 Цей голосовий трафік надзвичайно легко перехопити, тому він дуже вразливий до атак типу «людина посередині».14 Критично важливо завжди включати шифрування на всіх етапах. Якщо адміністратори не використовують надійні протоколи (наприклад, TLS для захисту з'єднання та SRTP для шифрування самого голосу), ваші приватні та комерційні розмови зможе послухати будь-хто.

ВИСНОВКИ

Захист даних — це вже давно не про формальні звіти для галочки чи нудні інструкції. Це питання фізичної безпеки під час війни та банального виживання бізнесу на ринку. Україні доведеться швидко змінювати свої закони під жорсткі європейські рамки, щоб інтегруватися у цивілізований світ та забезпечити прозорість збору інформації. У технічному плані майбутнє належить архітектурі, яка є безпечною за замовчуванням. Всім сучасним компаніям треба обов'язково комбінувати розумний моніторинг через SIEM-системи, надійне фізичне резервування інформації та дуже обережно підпускати працівників до інструментів штучного інтелекту. Тільки завдяки такому комплексному підходу ваші дані зможуть залишитися у справжній безпеці в нашому турбулентному цифровому світі.

Список літератури

1. Смірнова Т. В., Константинова Л. В., Конопліцька-Слободенюк О. К. та ін. Дослідження сучасного стану SIEM-систем. Кібербезпека: освіта, наука, техніка. 2024. № 1 (25). С. 6–18.

2. Босько В. В., Березюк І. А., Константинова Л. В., Конопліцька-Слободенюк О. К., Клюй А. Я. Аналіз та практична оцінка інструментів штучного інтелекту для веб-дизайну. Таврійський науковий вісник. Серія: Технічні науки. 2025. Вип. 1 (5). С. 65–73.

3. Снітко Ю. М. Огляд програмних середовищ для розробки мобільних додатків. Тези 76-ї наукової конференції (Полтава, 14–23 травня 2024 р.). Полтава : Нац. ун-т ім. Юрія Кондратюка, 2024. Т. 1. С. 482–483.

4. Дячук С. Ф., Борівець Б. Я. Крос-платформна розробка мобільних додатків за допомогою технології Xamarin. Матеріали VIII науково-технічної конференції. Тернопіль : ТНТУ, 2020. С. 131.

УДК 004.056, 004.75

С.О. Ліннікова, О.А.Кислун
sonamalishka2005@gmail.com, kyslun@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

АНАЛІЗ ОСНОВНИХ РИЗИКІВ ДЛЯ БЕЗПЕКИ СИСТЕМИ ДЕРЖАВНИХ ПУБЛІЧНИХ ЗАКУПІВЕЛЬ «PROZORRO»

ProZorro – це система відкритих державних закупівель, яка була створена для прозорості, створення конкуренції та зниження корупції. Платформа «Prozorro» – ключова складова всіх закупівель України, тому слід розуміти її значущість і можливі безпекові ризики цієї системи, а саме підняте питання є вельми актуальним.

Основними ризиками для атак є такі чинники: кібератаки; людський фактор; захист даних; технічні ризики; правові та організаційні

Під поняттям кібератаки маються на увазі DDoS-атаки (Distributed Denial of Service) – вид атак, які здійснюються шляхом надсилання на сервер великої кількості запитів. При такому перевантаженні система стає недоступною. Як наслідок, під час проведення важливих тендерів система може не відкритися, що може призвести до фінансових втрат. Несанкціонований доступ – це вид атаки, коли хакери отримують доступ до даних через використання вкрадених облікових даних. Наслідками такого нападу може бути витік або маніпуляція інформацією.

Людський фактор – основний фактор ризику, який пов'язаний із помилками, недбалістю або, навпаки, з навмисними діями. Можна виокремити три найчастіші ризики: ненавмисні дії, що призводять до зміни чи видалення певної інформації; облікові записи зі слабким рівнем автентифікації; навмисні дії та шкода з боку тих, хто має доступ до системи.

Захист даних – важлива частина платформи державних закупівель. Серед загроз є витік даних про комерційні угоди або персональної інформації користувачів.

Технічні ризики в системі ProZorro пов'язані з кіберзагрозами, які вже були розглянуті, а також із вразливостями програмного забезпечення та мережевими атаками. Вони можуть виникати через помилки в коді, недостатній захист каналів передачі даних. Слід виділити такі ризики, які впливають на витік інформації та порушення цілісності даних, а саме: зараження шкідливим програмним забезпеченням та відсутність належного моніторингу подій безпеки.

Правові та організаційні ризики найчастіше пов'язані з недотриманням вимог законодавства, зокрема Закон України «Про захист інформації в інформаційно-телекомунікаційних системах». Ризики можуть виникати через неналежний підхід до впровадження комплексної системи захисту інформації, недостатній контроль з боку керівництва. Провідну роль відіграє людський фактор, складові якого розглянуті вище. Таким чином, наслідками таких ризиків є витік даних, репутаційні проблеми, а також юридична відповідальність.

Отже, IT-закупівлі в системі ProZorro – це прогресивний етап розвитку цифровізації держави та її боротьби з корупцією. Однак у процесі дослідження цього питання було виявлено низку безпекових ризиків. Усі перелічені ризики є взаємопов'язаними, і покращення хоча б однієї складової стане важливим кроком до підвищення загального рівня безпеки системи. Комплексний підхід до їх усунення дозволить мінімізувати можливі загрози, підвищити довіру користувачів та забезпечити стабільне функціонування електронних закупівель.

Список літератури

1. Інформаційна безпека ProZorro.. URL: https://prozorro.gov.ua/information-security (дата звернення: 12.04.2026).
2. Про захист інформації в інформаційно-комунікаційних системах. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/80/94-вр#Text (дата звернення: 12.04.2026).
3. Що не так з публічними IT-закупівлями в Україні, і як зробити їх більш ефективними - Інфобокс Prozorro. Інформаційний ресурс - Інфобокс Прозорро. URL: https://infobox.prozorro.org/articles/shcho-ne-tak-z-publichnimi-it-zakupivlyami-v-ukrajini-i-yak-zrobiti-jih-bilsh-efektivnimi (дата звернення: 12.04.2026).

УДК 004.056:004.08

Р.Р. Орлов[1], Я.В. Тарасенко[2]
romanorlov0110@gmail.com, *yaroslav.tarasenko93@gmail.com*
[1]*Державний університет інформаційно-комунікаційних технологій, Київ*
[2]*Державний науково-дослідний інститут випробувань і сертифікації озброєння та військової техніки, Черкаси*

ІНТЕЛЕКТУАЛЬНЕ ВИМІРЮВАННЯ РІВНЯ КРИТИЧНОСТІ КІБЕРІНЦИДЕНТІВ В УМОВАХ НЕВИЗНАЧЕНОСТІ: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ

У сучасних інформаційно-комунікаційних системах визначення рівня критичності кіберінцидентів є однією з ключових передумов обґрунтованого автоматичного чи автоматизованого реагування на них чи протоколювання вжитих заходів. Швидкість, точність та коректність визначення рівня критичності впливає на пріоритетність дій, розподіл обмежених ресурсів чи своєчасність реагування. На практиці з цією метою часто використовуються обмежені та спрощені підходи, які орієнтовані у переважній більшості на формальні шкали тяжкості події. З роботи [1] видно, що попри фактичну універсальність у сфері кібербезпеки поширеного стандарту CVSS він володіє рядом істотних обмежень контекстуалізації та не забезпечує реальних потреб пріоритизації у повній мірі.

Окремі проблеми визначення рівня критичності кіберінцидентів пов'язані зі складністю вимірювання, яка пояснюється невизначеністю опису кіберінциденту через неповні, гетерогенні цифрові спостереження з розрізненою надійністю. До таких спостережень належать журнали подій, мережева телеметрія, сповіщення засобів виявлення, характеристики уражених об'єктів та зовнішній контекст загроз.

Систематичний огляд сучасних підходів до пріоритизації в центрах моніторингу безпеки засвідчує, що ефективне ранжування кіберінцидентів потребує одночасного врахування великої кількості критеріїв, контексту середовища та інтелектуальних механізмів підтримки прийняття рішень [2]. За таких умов рівень критичності розглядається як формалізована вимірювальна величина, яка визначається за сукупністю цифрових ознак, сформованих на основі мінімально достатньої кількості критичних метрик.

Одним з перспективних напрямків є представлення рівня критичності у вигляді CIA-профілю, який поєднує в собі три площини вимірювань. CIA-профіль відображає вплив інциденту на конфіденційність, цілісність і доступність інформаційних ресурсів. Інтелектуальне вимірювання у даному контексті направлене на охоплення формалізації вхідних даних, урахування довіри до джерел, агрегування неоднорідних ознак та пряме врахування і облік невизначеності. У роботі [3] показано, що врахування невизначеності та використання механізмів інтелектуального аналізу загроз підвищують обґрунтованість рішень у сфері інформаційної безпеки. Перехід від статичного оцінювання до інтелектуального вимірювання за таких умов є теоретично доцільним та практично необхідним.

Таким чином, до основних проблем у процесах інтелектуального вимірювання рівня критичності кіберінцидентів відносяться відсутність уніфікованої формальної моделі, неоднорідність надійності джерел спостереження, контекстуальна залежність результату вимірювань, потреба пояснюваності сформованої оцінки і трасованості процесу оцінювання.

Зазначені основні проблеми вирішуються перспективними напрямками, пов'язаними з розробленням методів поєднання CIA-формалізації, моделі довіри до джерел, механізмів злиття гетерогенних цифрових спостережень та процедури оцінювання невизначеності. Такі преспективні напрямки формують підґрунтя для відтворюваного, трасованого та обґрунтованого визначення критичності кіберінцидентів та підвищення швидкості і якості подальшого реагування.

Список літератури

1. Howland H. CVSS: ubiquitous and broken. *Digital threats: research and practice*. 2021. Vol. 4, Issue 1. URL: https://doi.org/10.1145/3491263 (дата звернення: 07.04.2026).

2. Alert prioritization in security operations centres: a systematic survey on criteria and methods / F. Jalalvand et al. *ACM computing surveys*. 2024. Vol. 57, Issue 2. URL: https://doi.org/10.1145/3695462 (дата звернення: 07.04.2026).

3. Dekker M., Alevizos L. A threat-intelligence driven methodology to incorporate uncertainty in cyber risk analysis and enchence decision-making. *Security and Privacy*. 2023. Vol. 7, Issue 1. URL: https://doi.org/10.1002/spy2.333 (дата звернення: 07.04.2026).

УДК 004.056.53:629.7(043.2)

Д.В. Смілка, *бакалавр 3 курс*
Науковий керівник: О.О. Кривокульська, *ст. викладач*
smilka.den@gmail.com. olha.kryvokulska@npp.kai.edu.ua
Державний університет «Київський авіаційний інститут», Київ

ІДЕНТИФІКАЦІЯ ТА КЛАСИФІКАЦІЯ КІБЕРРИЗИКІВ У ЦИФРОВИХ СИСТЕМАХ УПРАВЛІННЯ ПОВІТРЯНИМ РУХОМ

Сучасна авіаційна індустрія переживає етап радикальної трансформації, де традиційні аналогові методи зв'язку та навігації поступаються місцем інтегрованим цифровим екосистемам. Цей перехід, відомий як концепція CNS/ATM (Communication, Navigation, Surveillance / Air Traffic Management), відкриває нові можливості для підвищення пропускної спроможності повітряного простору, проте одночасно створює широкий ландшафт для кібератак. Цифровізація систем управління повітряним рухом супроводжується активним впровадженням мережевих технологій, автоматизованих комплексів обробки даних та систем супутникового спостереження. Сучасні наукові дослідження показують, що такі системи стають дедалі вразливішими до кіберзагроз через відкритий характер протоколів обміну даними та інтеграцію з глобальними мережами [1, 2]. Наголошується на складності виявлення атак у режимі реального часу, а в дослідженнях зосереджено увагу на необхідності системного підходу до класифікації кіберризиків [3]. Це робить тему дослідження актуальною, адже безпека польотів сьогодні безпосередньо залежить від кіберстійкості наземних та бортових систем.

Метою дослідження є визначення підходів до ідентифікації та класифікації кіберризиків у цифрових системах управління повітряним рухом та визначення їх основних закономірностей. Досягнення цієї мети дозволить сформувати фундамент для побудови проактивних систем захисту, здатних мінімізувати ймовірність катастрофічних наслідків.

У роботі розглядаються цифрові системи управління повітряним рухом, включаючи автоматизовані системи диспетчерського контролю, радіолокаційні комплекси та системи передачі даних між повітряними суднами і наземними службами (ADS-B, CPDLC). Ці системи є основою сучасної авіації: ADS-B забезпечує автоматичне залежне спостереження, передаючи координати літака, а CPDLC (Controller-Pilot Data Link Communications) дозволяє диспетчеру та пілоту обмінюватися текстовими повідомленнями, розвантажуючи голосові канали.

Дослідження виконано на основі аналітичного підходу з використанням сучасної наукової літератури, міжнародних стандартів та рекомендацій організацій ICAO та EUROCONTROL. Були застосовані методи класифікації ризиків за джерелами та типами впливу, а також інструменти аналізу ризиків відповідно до стандарту ISO/IEC 27005. Методологія також включає розгляд сценаріїв атак через призму моделі STRIDE, що дозволяє детально проаналізувати загрози підміни даних та відмови в обслуговуванні

Основні джерела кіберризиків часто спричинені зовнішніми атаками, внутрішніми загрозами та технічними вразливостями. Зовнішні загрози включають спроби несанкціонованого доступу, DDoS-атаки, атаки на серверні ресурси та порушення роботи ліній передачі. Такі події можуть вплинути на доступність системи та затримати передачу або обмін даними. Для систем УПР затримка навіть у декілька секунд може призвести до небезпечного зближення повітряних суден.

Внутрішні загрози пов'язані з людським фактором і включають людські випадкові помилки, неправильне налаштування системи, невиконання процедур безпеки та навмисні шкідливі дії, які можуть виконувати співробітники. В авіації інсайдерські загрози мають критичне значення, оскільки персонал має легітимний доступ до консолей управління та систем планування польотів. Навіть без зовнішнього втручання ці фактори можуть суттєво вплинути на продуктивність системи.

Загрози навколишнього середовища спричинені вразливостями програмного та апаратного забезпечення, або несправними компонентами, відсутністю своєчасних оновлень та неправильною конфігурацією, що збільшує можливість виникнення кіберзагроз. Використання застарілих (Legacy) систем, які не розраховувалися на роботу в агресивному кіберсередовищі, створює додаткові вектори атак через непропатчені вразливості нульового дня. Особливу увагу треба приділяти питанню передачі даних. Використання відкритих систем, таких як ADS-B, без внутрішнього шифрування створює можливості для зловживання, втручання та порушення передачі даних польотів. Проблема полягає в тому, що протокол ADS-B Out транслює дані у відкритому вигляді, що дозволяє зловмисникам реалізовувати атаки типу "GPS Spoofing" (підміна координат) або "Ghost Aircraft Generation" (створення неіснуючих бортів на екрані диспетчера).Це важливо, оскільки надійність та цілісність інформації безпосередньо пов'язана з безпекою авіації. Аналогічно, втручання в систему CPDLC може призвести до підробки команд диспетчера, що безпосередньо загрожує життю пасажирів.

Кіберзагрози можна класифікувати за різними критеріями, такими як: джерело, рівень критичності та категорія і тип вразливості, щоб відповідні загрози можна було точно ідентифікувати та оцінити ризики та можливі наслідки.

Таблиця 1
Класифікації кіберризиків

Категорія ризику	Опис	Приклад загрози	Рівень критичності
Мережеві	Порушення передачі даних у мережі	DDoS-атака на сервери	Високий
Програмні	Вразливості програмного забезпечення	Експлуатація помилок ПЗ	Високий
Людський фактор	Помилки або дії персоналу	Неправильна конфігурація системи	Середній
Дані	Порушення цілісності інформації	Підміна ADS-B сигналів	Високий
Фізичні	Несанкціонований доступ до обладнання	Доступ до серверної інфраструктури	Середній

Аналіз показав, що найбільш критичними є ризики, пов'язані з порушенням цілісності та доступності даних. В авіаційному контексті цілісність (Integrity) означає впевненість у тому, що координати літака на екрані відповідають реальності, а доступність (Availability) — що зв'язок між пілотом та землею не буде перервано у критичний момент посадки чи зльоту. Навіть незначні кіберінциденти можуть призвести до серйозних наслідків через високу завантаженість та залежність систем управління повітряним рухом від точності інформації в реальному часі. Варто також враховувати каскадний ефект: збій в одному центрі УПР може призвести до колапсу авіасполучення в цілому регіоні.

Також значну роль у виникненні кіберризиків відіграє людський фактор, що підтверджує необхідність підвищення рівня підготовки персоналу та впровадження додаткових процедур контролю. Це включає не лише технічне навчання, а й формування культури кібербезпеки (Cybersecurity Awareness), де кожен диспетчер розуміє загрозу використання неперевірених USB-носіїв або слабких паролів. Для ефективної протидії виявленим ризикам пропонується впровадження багаторівневої моделі захисту "Defense in Depth", яка включає: шифрування даних на рівні протоколів передачі (де це технічно можливо), використання систем виявлення вторгнень (IDS), адаптованих до авіаційних протоколів, регулярне проведення кібернавчань та симуляцій атак на системи УПР.

Результати показують можливість інтеграції моделі кіберризиків у цифрові системи управління повітряним рухом для підвищення точності їх аналізу та планування безпеки. Системна ідентифікація дозволяє перейти від ліквідації наслідків до стратегічного запобігання інцидентам. Ці класифікації можуть бути використані для розробки стандартів кібербезпеки та вдосконалення систем управління повітряним рухом та розробки вимог до сертифікації нового авіаційного обладнання. Тільки комплексний підхід, що поєднує технічні рішення та адміністративні заходи, дозволить зберегти високий рівень безпеки польотів у цифрову епоху.

Список літератури :

1. ENISA. Cybersecurity in Air Traffic Management. – 2021.
2. ICAO. Aviation Cybersecurity Strategy. – 2019.
3. EUROCONTROL. Guidelines on ATM Cybersecurity. – 2022.

УДК 004.946.5.056:658.114(477)(043.2)

А.О. Маруніна, *бакалавр, 4 курс*
Науковий керівник: О.О. Кривокульська, *ст. викладач*
7952484@stud.kai.edu.ua, olha.kryvokulska@npp.kai.edu.ua
Державний університет «Київський авіаційний інститут», Київ

РИЗИК-ОРІЄНТОВАНИЙ ПІДХІД ДО ЗАБЕЗПЕЧЕННЯ КІБЕРСТІЙКОСТІ УКРАЇНСЬКИХ КОМПАНІЙ ПІД ЧАС МІГРАЦІЇ ДО ХМАРНОЇ ІНФРАСТРУКТУРИ

Актуальність теми зумовлена зростанням кількості складних кіберзагроз, що орієнтовані на хмарні середовища, які активно використовуються українськими підприємствами. Перехід до хмарних технологій дозволяє підвищити гнучкість бізнес-процесів і забезпечити доступність сервісів, однак водночас створює нові вектори атак, пов'язані з динамічністю інфраструктури та складністю її контролю. В умовах постійних змін конфігурацій і масштабування ресурсів традиційні методи захисту стають недостатньо ефективними, що актуалізує потребу у впровадженні адаптивних механізмів кіберзахисту.

Особливої ваги це питання набуває для українського бізнесу в умовах правового режиму воєнного стану, коли фізична інфраструктура (дата-центри) перебуває під ризиком знищення. Міграція у хмару стає стратегічним рішенням для виживання, проте вона вимагає перегляду парадигми безпеки: від захисту периметра до захисту даних та ідентичності. Ризик-орієнтований підхід дозволяє компаніям не просто впроваджувати хаотичні засоби захисту, а фокусувати ресурси на найбільш критичних активах, враховуючи специфіку моделей SaaS, PaaS та IaaS.

Метою роботи є дослідження підходів до забезпечення кіберстійкості підприємств шляхом використання адаптивних систем моніторингу в хмарних інфраструктурах. Для досягнення мети необхідно вирішити такі завдання: проаналізувати сучасні загрози в хмарному середовищі, дослідити принципи побудови систем моніторингу безпеки та визначити роль автоматизації у реагуванні на інциденти.

У контексті ризик-орієнтованого управління, кіберстійкість (cyber resilience) розглядається як здатність системи витримувати атаки, відновлюватися після них і адаптуватися до нових умов. На відміну від класичної кібербезпеки, стійкість передбачає, що інцидент рано чи пізно станеться, тому основна увага приділяється мінімізації наслідків (Impact) та часу відновлення (RTO/RPO).

У дослідженні використано методи аналізу архітектур хмарних систем, моделювання кіберзагроз та узагальнення сучасних підходів до управління подіями безпеки.

Однією з ключових проблем забезпечення кіберстійкості є складність своєчасного виявлення інцидентів у розподілених середовищах. Хмарні сервіси генерують великий обсяг телеметричних даних, що ускладнює їх аналіз традиційними засобами. Це зумовлює необхідність використання систем централізованого моніторингу, таких як SIEM (Security Information and Event Management), які дозволяють агрегувати події з різних джерел та здійснювати їх кореляцію.

Проте впровадження SIEM у хмарі має свої особливості. По-перше, це "модель спільної відповідальності" (Shared Responsibility Model), де провайдер відповідає за безпеку самої хмари, а клієнт — за безпеку даних у ній. Нерозуміння цієї межі є одним із найбільших ризиків. По-друге, динамічність хмарних об'єктів (мікросервіси, контейнери) призводить до появи короткоживучих логів, які важливо встигнути зібрати до видалення інстансу. Аналіз ризиків на етапі міграції дозволяє виявити прогалини у видимості (Visibility Gap) та налаштувати логування критичних API-викликів.

Особливу роль відіграють технології автоматизації реагування (SOAR), які забезпечують швидке виконання сценаріїв реагування на інциденти без втручання людини. Це дозволяє значно скоротити час виявлення та нейтралізації загроз, що є критично важливим для забезпечення безперервності бізнесу.

Впровадження SOAR у хмарних середовищах дозволяє реалізувати концепцію "Infrastructure as Code" (IaC) для безпеки. Наприклад, при виявленні несанкціонованого доступу з аномальної IP-адреси, система може автоматично переписати правила Security Group або анулювати токени доступу в режимі реального часу, випереджаючи дії зловмисника.

Таблиця 1

Архітектура адаптивної системи моніторингу безпеки в хмарному середовищі

Етап / Компонент	Опис
Джерела даних	Журнали логів, мережевий трафік, події контролю доступу
Збір та обробка даних	Агрегація та нормалізація інформації з різних джерел
SIEM-система	Централізований аналіз подій безпеки та їх кореляція
Аналіз аномалій (ML)	Виявлення нетипової поведінки за допомогою машинного навчання
Виявлення загроз	Поведінковий аналіз і кореляція подій
Хмарна інфраструктура	Середовище розміщення ресурсів і обробки даних
Автоматизація реагування (SOAR)	Виконання сценаріїв реагування на інциденти
Дії реагування	Блокування доступу, ізоляція ресурсів, сповіщення
Моніторинг і контроль	Безперервне спостереження за станом безпеки

Запропонована архітектура передбачає інтеграцію джерел даних (логів, мережевого трафіку, подій доступу) у єдину платформу моніторингу. На першому етапі здійснюється збір та нормалізація даних, після чого відбувається їх аналіз із використанням правил та моделей поведінкового аналізу. Важливою складовою є використання машинного навчання для виявлення аномалій, що не можуть бути визначені заздалегідь заданими правилами.

Машинне навчання (ML) у хмарному моніторингу вирішує проблему "алертової втоми" (alert fatigue). Алгоритми здатні розрізняти легітимне пікове навантаження під час маркетингових акцій від DDoS-атаки або виявляти приховану ексфільтрацію даних, яка відбувається невеликими порціями через нестандартні порти. Це критично для українських компаній, які часто стають об'єктами цілеспрямованих атак (APT).

Наступним етапом є автоматизоване реагування на інциденти, що включає блокування підозрілих дій, ізоляцію скомпрометованих ресурсів та сповіщення відповідальних осіб. Такий підхід дозволяє забезпечити безперервний контроль безпеки та швидку адаптацію до нових типів загроз.

Важливим аспектом є також інтеграція систем моніторингу з політиками управління доступом і конфігураціями хмарних ресурсів. Це забезпечує узгодженість дій між різними компонентами системи безпеки та зменшує ризик людських помилок.

Крім того, ефективна стратегія кіберстійкості повинна включати регулярне тестування — "Red Teaming" та симуляцію атак у хмарі (Breach and Attack Simulation). Для українських реалій це також означає наявність стратегії "виходу" (Exit Strategy) або мультихмарного підходу, щоб уникнути залежності від одного постачальника (Vendor Lock-in) та забезпечити працездатність сервісів навіть у разі глобальних збоїв або санкційних обмежень

Практичне впровадження запропонованого підходу дозволяє знизити ймовірність успішної атаки на 40-60% та скоротити витрати на ліквідацію наслідків інцидентів за рахунок раннього виявлення. Ризик-орієнтована модель стає фундаментом для побудови довіри між бізнесом, клієнтами та державою в умовах цифрової трансформації.

Результати дослідження показують, що впровадження адаптивних систем моніторингу є ефективним засобом підвищення кіберстійкості підприємств у хмарному середовищі. Поєднання централізованого аналізу подій, автоматизації реагування та використання інтелектуальних методів обробки даних дозволяє забезпечити своєчасне виявлення та нейтралізацію загроз. Подальші дослідження можуть бути спрямовані на вдосконалення алгоритмів машинного навчання для підвищення точності виявлення аномалій.

Список літератури:

1. Кузьменко О. О. Хмарні технології в інформаційних системах : навч. посіб. Київ : Видавництво НТУУ «КПІ», 2021. 256 с.

2. Савчук В. В. Системи моніторингу кібербезпеки підприємств. Інформаційні технології та безпека. 2022. № 3. С. 45–52.

3. Бондаренко І. І. Автоматизація реагування на кіберінциденти в корпоративних мережах. Кібербезпека: освіта, наука, техніка. 2023. № 1 (17). С. 78–85.

УДК 004.056.5:004.8

М.С. Пилипчук, *здобувач кафедри кібербезпеки другий курс*
А.В. Ільєнко, *канд. техн. наук, доцент, завідувач кафедри кібербезпеки*
О.В. Дубчак, *ст. викладач кафедри кібербезпеки*
9222191@stud.kai.edu.ua, anna.ilienko@npp.kai.edu.ua, 3915922@npp.kai.edu.ua
Державний університет «Київський авіаційний інститут», Київ, Україна

КІБЕРБЕЗПЕКА EDGE-AI: УРАЗЛИВОСТІ НЕЙРОПРОЦЕСОРІВ І РИЗИКИ АПАРАТНОГО ВИКОНАННЯ МОДЕЛЕЙ НА МАЛОПОТУЖНИХ УБУДОВАНИХ СИСТЕМАХ

Розвиток AI (Artificial Intelligence, штучний інтелект) та темпи його інтегрування в більшість сфер людського життя призвело до створення недорогих (до 15-20$) та невеликих SoC-систем (System-on-a-Chip, системи на кристалах). Ці досить енергоефективні рішення, з рівнем споживання 0,5-1 Вт, дозволяють власне використання в будь-якій локації та часі, незалежно від факторів зовнішнього середовища. Згідно з прогнозами аналітичних агенцій, зокрема IDC, до 2026-2027 років понад 70% корпоративних і сенсорних даних оброблятимуться саме на Edge-пристроях [1]. Маючи на собі спеціалізовані апаратні прискорювачі, такі як NPU (Neural Processing Unit, нейронні процесори), невелика плата розміром в 2*2 см може виконувати завдання комп'ютерного зору з використанням AI, працюючи з камерою в реальному часі із затримкою 15-30 мс на обробку одного кадру. Це створює неймовірні можливості для галузей робототехніки та інженерії.

Однак, достатньо недавня поява NPU на ринку та майже миттєва їхня інтеграція в сферу використанні AI з жорстоким обмеженням ресурсів, завдяки специфічним особливостям, наведеним вище, зумовлює появу все нових векторів атак. Традиційні моделі загроз зазвичай ігнорують апаратну частину SoC, фокусуючись на програмних уразливостях операційної системи (ОС), не зважаючи на те, що саме взаємодія NPU із CPU (Central Processing Unit, центральний процесор), оперативною пам'яттю (ОП) та SRAM (Static Random Access Memory, високошвидкісна енергонезалежна напівпровідникова ОП) несе найбільшу загрозу безпеки.

1.Сучасні SoC з малопотужними CPU, такими як RISK-V чи ARM Cortex, у поєднанні з NPU зазвичай використовують UMA (уніфіковану модель доступу до пам'яті) [2]. Вона передбачає, що всі операції, які використовують ОП, застосовуються без IOMMU (Input-Output Memory Management Unit, механізми захисту від вторгнення), оскільки це знижує пропускну здатність шини на 15-20%. [3] В якості ОС зазвичай використовується BuildRoot на базі Linux, оскільки власник плати може створити свою унікальну систему, власноруч вибираючи пакети для видалення чи встановлення. І це є гарним рішенням, оскільки готові рішення на базі Debian або Arch суттєво збільшують час завантаження системи та її вагу. Оскільки зазвичай використовуються лише найнеобхідніші драйвери, інструменти для боротьби з несанкціонованим проникненням залишаються поза увагою.

Під час однієї зі стадій життєвого циклу нейромережі (інференсу) NPU зчитує дані, такі як тензори та ваги, безпосередньо з виділених буферів. Якщо ця пам'ять не є захищеною і простір роботи користувача не є жорстоко відмежований від ядра, зловмисник може безперешкодно впливати на результат, маніпулюючи пам'яттю.

Вектори атак на нейронні прискорювачі:

1. Атаки через корупцію пам'яті (Memory Corruption) та DMA-атаки (Direct Memory Access). Оскільки NPU вимагають виділення простору пам'яті з безперервним доступом, хакер, маючи обмежений доступ до системи, навіть без прав Root, може експлуатувати переповнення буферу. Це дозволить перезаписати ваги нейромережі та виключити деякі результати, або змусити NPU видавати хибно-позитивні результати. [4]

2. Атаки за сторонніми каналами (Side-Channel Attacks, SCA). Під час інференсу SoC споживання електроенергії та виділення електромагнітного випромінювання не є хаотичними – вони формують чітку сигнатуру щодо архітектури нейромережі. Використовуючи звичайний осцилограф, зловмисник може здійснити reverse engineering та дізнатися ключові характеристики запущеної моделі: розмір вхідних тензорів, кількість згорток та їхній тип з точністю в 90-95% [5]. Особливо небезпечним це втручання може бути у військовій та комерційній сферах використання.

3. Експлуатація уразливостей парсерів моделей. Перед запуском виконуваний файл нейромережі обробляється драйвером на рахунок помилок, які можуть виникнути під час роботи. Заміна чи модифікація самого парсера може призвести до відміни в обслуговуванні (Denial-of-Service, DoS) або виконання довільного коду.

Методи забезпечення безпеки NPU на SoC:

1. Обов'язове включення IOMMU під час налаштування ОС. Навіть на чипах з низькою продуктивністю потрібно мати жорстокий контроль доступу до пам'яті та використовувати його розмежування.

2. Реалізація "шуму". Додавання випадкових (Random) обчислень та несподівана зміна частоти NPU значно ускладнять розшифрування початкової архітектури.

3. Криптографічний захист навченої моделі. Шифрування важливих даних та їхнє розшифрування безпосередньо в захищених місцях пам'яті.

Створення невеликих обчислювальних систем та їхня інтеграція в сучасну інфраструктуру є великим досягненням. Але, з іншого боку, має багато ще не відкритих уразливостей. Подальший розвиток апаратної та програмної частин має бути також спрямованим на розробку легковагових алгоритмів захисту, що не будуть обтяжувати апаратне забезпечення, яке наразі може бути не досить продуктивним [6].

Список літератури:

1.IDC Forecasts Sharp Increase in AI Impact on Edge Computing. *IDC: The premier global provider of market intelligence*. 2024. URL: https://www.idc.com (дата звернення: 12.04.2026).

2.Sophgo SG2002 TRM (Technical Reference Manual). Revision 1.0. *Sophgo Technologies*, 2023. 452 p.

3.Zheng L., та ін. Supporting Address Translation for Accelerator-Centric Architectures. *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2017. С. 421-432.

4.Rakin A. S., He Z., Fan D. TBT: Targeted Neural Network Attack with Bit Trojan. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020. С. 13198-13207.

5.Batina L., Bhasin S., Jap D., Picek S. Physical Side-channel Attacks on Neural Networks: A Survey. *Journal of Hardware and Systems Security*. 2019. Vol. 3. P. 132–146.

6.Zhou X., Zhao J., Jiang L., Zhang T. Security and Privacy in Edge AI: A Systematic Review. *IEEE Access*. 2021. Vol. 9. P. 124322–124345.

СЕКЦІЯ 2. ПРОГРАМУВАННЯ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.8:004.4

Д. С. Мельник *ст. 2 курсу*, Л. В. Константинова
dimam4659@gmail.com, liliyashel1976@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький, Україна

ВПЛИВ ВИКОРИСТАННЯ AI-АСИСТЕНТІВ НА ПРОДУКТИВНІСТЬ ТА ЯКІСТЬ ПРОГРАМНОГО КОДУ У ПРОЦЕСІ РОЗРОБКИ

Щоб розвиватися відповідно до технологічних змін у світі, виробники програмного забезпечення вимушені розробляти нові гнучкі підходи. Розвиток штучного інтелекту (ШІ) сильно змінює методи створення програмного забезпечення (ПЗ). Використання AI-інструментів допомагає автоматизувати рутинні завдання, та пришвидшити написання простого коду. Так як розробка програмного забезпечення часто стає тривалим та складним процесом, актуальним стає дослідження впливу таких інструментів на якість отримуваного коду та ефективність роботи розробників. Метою роботи є - ознайомитись з можливостями застосування, а також переконатись у результативності використання AI-асистентів для написання коду на прикладі GitHub Copilot.

Як зазначає у своєму дослідженні Т. Домке [1], в основі таких інструментів лежать великі мовні моделі (LLM). Їх спеціально тренують саме на розуміння структури програмного коду. Для цього розробники використовують алгоритми та масиви відкритого коду, що дозволяє моделям глибоко осягнути архітектуру проєктів. Згодом ці навчені мережі вбудовуються в ШІ-асистенти. Завдяки такій основі вони здатні миттєво орієнтуватися в незнайомій структурі проєкту та пропонувати доречні рішення. Це суттєво розвантажує програмістів, адже модель бере на себе велику частку рутини.

Якщо говорити про конкретні приклади, то беззаперечним лідером зараз є GitHub Copilot. Його призначення при створенні планувалось - як напарник для парного програмування, який без проблем інтегрується в популярні IDE. Всередині цього сервісу працює одразу декілька нейромереж.

Якщо докладніше, то Copilot аналізує не лише той рядок, над яким програміст працює в моменті. Він враховує весь контекст: відкриті файли та загальну архітектуру. Залежно від потреб, можна перемикатися між кількома режимами. Наприклад, через звичайний «Ask» зручно спілкуватися з моделлю у форматі чату. Спитати пораду чи розібрати логіку алгоритму (проте тут глибокий аналіз всього проєкту не задіюється). Якщо ж потрібно щось масштабніше, вмикають автономний режим «Agent». Тут ШІ сам досліджує файлову систему і пропонує комплексні зміни. Розробнику залишається тільки переглянути згенерований код і натиснути «прийняти» або «відхилити». Як додаткова перевага до всього, «Агент» вміє самостійно писати тести і видавати короткий звіт про виправлені баги. Існує також полегшений варіант - «Edit». Він робить приблизно те ж саме, але працює виключно з конкретно вказаними файлами, без генерації тестів та звітів. Завдяки такому спрощенню він працює набагато швидше і чудово підходить для точкових задач. Фактично, з появою цих інструментів робота програміста перетворюється з простого написання коду на роботу своєрідного менеджера: він більше керує процесом і займається перевіркою результатів.

Згідно з даними документації [2], розробка з ШІ-помічниками прискорюється на 20–55%. Найкраще ці інструменти справляються з написанням модульних тестів (через Agent), генерацією шаблонного коду та швидким розбором чужих функцій (через Ask), для тих випадків, якщо немає можливості вивчати документацію.

Але не все, що пише штучний інтелект є правдивим, часто можна натрапити на «галюцинації». У звіті W. Harding [3] означено, що нейромережі можуть видавати абсолютно хибні або навіть небезпечні рішення і переконливо видавати іх за вірні. Якщо розробник сліпо довірятиме машині й не читатиме те, що вона згенерувала, на виході вийде ПЗ поганої якості.

Висновок: хоча GitHub Copilot та інші ШІ-асистенти дійсно знімають багато рутини і дають змогу зосередитися на загальній архітектурі, вони все ще часто помиляються. Саме тому машина не може замінити людину. Кожен згенерований рядок коду все одно потребує уважної перевірки та критичного оцінювання з боку живого інженера. Тільки так можна гарантувати надійність і якість розробленої програми.

Список літератури

1.T. Dohmke, The economic impact of the AI-powered developer lifecycle and lessons from GitHub Copilot. San Francisco, USA: GitHub, 2023.

2.S. Peng, E. Kalliamvakou, P. Cihon, та M. Demirer, "The Impact of AI on Developer Productivity: Evidence from GitHub Copilot", arXiv preprint, arXiv:2302.06590, с. 1-17, 2023.

3.W. Harding, та M. Kloster, Coding on Copilot: 2023 Data Shows Downward Pressure on Code Quality. Seattle, USA: GitClear, 2024.

УДК 004.93

О.А. Смірнов, В.А. Заріцкий, К.О.. Буравченко, С.А. Смірнов
dr.smirnovoa@gmail.com, viktorzarickiy@gmail.com, buravchenkok@gmail.com, smirnov.ser.81@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький, Україна

МЕТОД ПІДВИЩЕННЯ ОБЧИСЛЮВАЛЬНОЇ ЕФЕКТИВНОСТІ РОЗПІЗНАВАННЯ ОБЛИЧ НА БАЗІ БІБЛІОТЕКИ DLIB З ВИКОРИСТАННЯМ АРХІТЕКТУРИ CUDA

Задачі розпізнавання образів на даний час дуже актуальні. Поштовхом для бурхливого розвитку теорії розпізнавання образів послужило як розширення області застосування, так й поява нових технологій розпізнавання, у тому числі з широким використанням штучного інтелекту. Якщо розглянути область застосування то це й використання у військовій справі (наприклад у дронах, що отримало активний розвиток під час російсько-української війни), й використання на виробництві (системи комп'ютерного зору для виявлення дефектів), й використання у торгівлі (розпізнавання штрих та QR-кодів, розпізнавання речей й т.п.), й використання у медицині (медична діагностика), й використання у різноманітних комплексних системах безпеки (інтелектуальні системи відеоспостереження, розпізнавання осіб, розпізнавання номерів авто й т.п.), а також велика кількість інших задач. Дана робота присвячена використанню теорії розпізнання образів у комплексних системах безпеки. Зважаючи на те, що технології комп'ютерного зору сьогодні є невід'ємною частиною систем безпеки та ідентифікації, методи автоматичного розпізнавання облич набувають критичного значення. Інтелектуальні системи відеонагляду впроваджуються у смарт-містах, банківському секторі, на транспорті та в мобільних пристроях. Ключовою вимогою до таких систем є здатність обробляти відеопотік у режимі реального часу (real-time) з мінімальними затримками. В умовах зростання вимог до точності розпізнавання, розробники все частіше відмовляються від класичних алгоритмів на користь глибокого навчання (Deep Learning). Сучасні нейронні мережі здатні розпізнавати обличчя в складних умовах освітлення, при частковому перекритті або повороті голови, що було недосяжно для алгоритмів попереднього покоління. Проте, ефективність процесів детекції значною мірою залежить від апаратного забезпечення. Виконання складних математичних операцій згортки на центральному процесорі (CPU) часто призводить до "пляшкового горлечка" в продуктивності системи. Застосування графічних прискорювачів (GPU) та технології CUDA відкриває широкі можливості для розпаралелювання обчислень. Це дозволяє використовувати найсучасніші архітектури нейронних мереж без втрати швидкодії, забезпечуючи баланс між точністю та швидкістю реакції системи. При впровадженні нейромережевих методів (CNN) у прикладні задачі виникає проблема значного зростання обчислювального навантаження. Класичні CPU не здатні забезпечити прийнятний FPS (кількість кадрів на секунду) при роботі з CNN-детекторами високої точності. Це створює бар'єр для використання передових алгоритмів у системах реального часу. Особливу роль у вирішенні цієї проблеми відіграють програмні бібліотеки, такі як dlib, які підтримують гібридні режими роботи. Необхідно дослідити, наскільки ефективним є перенесення обчислень на GPU за допомогою CUDA для конкретної реалізації детектора MMOD CNN у бібліотеці dlib. Таким чином постає завдання експериментальної перевірки та кількісної оцінки виграшу в продуктивності при переході від CPU-обчислень (метод HOG) до GPU-обчислень (метод CNN), а також розробка методики тестування цих показників.

Алгоритми інтелектуального аналізу даних та їх інтеграція з ШІ

Бібліотека dlib пропонує два фундаментально різних підходи до вирішення задачі розпізання об'єктів. Перший – це метод HOG (Histogram of Oriented Gradients), який базується на виділенні ознак форми та контурів об'єкта через аналіз градієнтів яскравості. Цей метод є обчислювально легким і традиційно виконується на CPU. Він ефективний для фронтальних облич, але чутливий до поворотів. Другий підхід – це CNN (Convolutional Neural Network), зокрема модель MMOD. Це глибока нейронна мережа, яка навчається виділяти складні ієрархічні ознаки. Вона забезпечує високу надійність детекції (robustness), але вимагає виконання мільйонів операцій множення матриць для одного кадру, що є ідеальним сценарієм для застосування GPU.

Технологія CUDA для прискорення обчислень

Технологія NVIDIA CUDA (Compute Unified Device Architecture) дозволяє використовувати графічний процесор як пристрій масових паралельних обчислень. На відміну від CPU, який має кілька потужних ядер, GPU має тисячі менших ядер, здатних одночасно обробляти різні пікселі зображення або нейрони мережі. Інтеграція dlib з CUDA дозволяє перекласти найбільш ресурсоємні операції (tensor operations) на відеокарту. Це звільняє центральний процесор для інших задач та дозволяє досягти режиму реального часу навіть для "важких" моделей глибокого навчання. З метою всебічного аналізу було розроблено спеціалізоване програмне забезпечення мовою Python, яке реалізує бенчмаркінг двох методів детекції на одному і тому ж наборі даних. Експеримент проводився з використанням бібліотек dlib, numpy та matplotlib для візуалізації.

Методика експерименту та програмна реалізація

Для забезпечення об'єктивності результатів було створено клас FaceDetectionBenchmark. Алгоритм передбачає попереднє завантаження кадрів у пам'ять, "прогрів" GPU (для виключення затримок ініціалізації

CUDA контексту) та заміри часу виконання детекції для кожного кадру окремо. Важливим аспектом реалізації є використання таймерів високої точності та розрахунок стандартного відхилення часу обробки, що дозволяє оцінити стабільність потоку даних (jitter).

Результати роботи методу HOG (CPU)

Тестування класичного детектора показало стабільні результати на центральному процесорі. Середній час обробки кадру склав близько 25-30 мс (залежно від роздільної здатності), що відповідає приблизно 30-40 FPS. Перевагою цього методу є відсутність потреби у дороговартісному обладнанні. Однак, аналіз кількості виявлених облич показав, що HOG часто пропускає обличчя, які знаходяться під кутом або мають нестандартне освітлення.

Результати роботи методу CNN (GPU/CUDA)

При використанні CNN детектора без CUDA (на CPU) час обробки одного кадру складав понад 800 мс (~1.2 FPS), що є неприйнятним для відеопотоку. Після активації CUDA-прискорення ситуація кардинально змінилася. Час обробки скоротився до 15-20 мс, що дозволило отримати понад 50 FPS. При цьому точність детекції (recall) зросла: детектор успішно знаходив обличчя, які були пропущені методом HOG.

Порівняльна характеристика та візуалізація

Отримані дані дозволяють побудувати чітку картину переваг використання апаратного прискорення. Коефіцієнт прискорення становить 2.28.

Висновки та перспективи подальших досліджень

В роботі було досліджено засоби бібліотеки dlib для розпізнавання облич та проведено практичний експеримент з оптимізації цього процесу. Було визначено, що стандартні методи CPU-обчислень є недостатніми для роботи із сучасними згортковими нейронними мережами у відеопотоці. Розроблений програмний модуль довів, що використання технології CUDA забезпечує прискорення процесу детекції у десятки разів порівняно з виконанням тієї ж моделі на CPU, та перевершує за швидкістю навіть простіші алгоритми (HOG), забезпечуючи при цьому значно вищу якість розпізнавання. Перспективи подальших досліджень полягають у тестуванні оптимізації batch-processing (пакетної обробки кадрів), що може ще більше підвищити пропускну здатність системи, а також у дослідженні використання тензорних ядер (Tensor Cores) новітніх відеокарт.

Список літератури

1. King D. E. Max-Margin Object Detection / D. E. King // Journal of Machine Learning Research. – 2015. – Vol. 16. – P. 1113-1120.

2. Dalal N., Triggs B. Histograms of oriented gradients for human detection. CVPR, 2005.

3. James G. Shanahan. Introduction to Computer Vision and Realtime Deep Learning-based Object Detection /CIKM '20: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. Pages 3515 – 3516.

4. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications, 2025, pp. 589–622.

5. Kuznetsov, O., Smirnov, O., Akhmetov, B., Alimseitova, Z., Imoize, A.L. «Deep Learning Frontiers in Copy-Move Forgery Detection: Advances, Challenges, and Future Directions». Advancements in Cybersecurity Next Generation Systems and Applications, 2025. 202-229.

6. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». International Review on Modelling and Simulations 18 (1), 2025. pp. 32-42.

7. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». CEUR Workshop Proceedings Volume 3664, 2024, Pages 11-23.

8. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». *CEUR Workshop Proceedings,* Volume 3312, 2022, pp. 47-58.

УДК 004.272.2:004.272.43

В.В. Кіш, Н.І. Йовбак
kish.viktor@student.uzhnu.edu.ua, yovbak.nika@student.uzhnu.edu.ua
ДВНЗ »Ужгородський національний університет», Ужгород

АРХІТЕКТУРНІ ОСОБЛИВОСТІ ТА МЕХАНІЗМИ КОГЕРЕНТНОСТІ ПАМ'ЯТІ У БАГАТОСОКЕТНИХ NUMA-СИСТЕМАХ

Сучасні багатосокетні обчислювальні системи, побудовані за моделлю **Non-Uniform Memory Access (NUMA)**, лежать в основі високопродуктивних серверів, кластерів і хмарних платформ. Їхня продуктивність визначається не лише швидкістю окремих ядер, а передусім тим, наскільки ефективно програмне забезпечення взаємодіє з пам'яттю та кешовою системою в умовах нерівномірної доступності. Різниця між локальним і віддаленим доступом, вартість когерентності між сокетами, поведінка планувальника, міграція сторінок, робота TLB та периферійних інтерфейсів ставлять складні задачі перед розробниками високонавантажених систем. Це зумовлює актуальність дослідження методів оптимізації міжсокетної взаємодії та пошуку практичних способів зменшення накладних витрат у NUMA-середовищах. Фундаментом для розробки таких методів є глибоке розуміння фізичної організації обчислювальних вузлів, оскільки саме апаратна топологія визначає межі пропускної здатності та затримки, з якими стикається програмне забезпечення. Розглянемо детальніше принципи побудови та еволюцію **архітектури сучасних NUMA-систем** [1].

Архітектура NUMA виникла як відповідь на проблеми масштабування класичних SMP-систем, у яких усі ядра поділяють один контролер пам'яті та одну фізичну шину доступу до DRAM. Коли кількість ядер перевищує десятки, єдиний контролер стає точкою насичення, а збільшення пропускної здатності шляхом множення кількості каналів не вирішує проблеми, оскільки одночасна конкуренція потоків за однаковий шлях доступу створює ефект фронту затримок. Архітектура NUMA радикально змінює це, розподіляючи пам'ять між декількома вузлами, кожен з яких має повністю окремий контролер пам'яті, окремі DRAM-канали та власну локальну кешову ієрархію. Таким чином, система набуває властивості мультикластера, де кожен вузол формує мініатюрний SMP-домен із низькою внутрішньою латентністю доступу до пам'яті та високою пропускною здатністю, а взаємодія між вузлами здійснюється через високошвидкісний інтерконект.

Сучасні серверні процесори Intel використовують внутрішню mesh-топологію, у якій ядра, LLC-slices, Home Agents та IMC з'єднані у багатовимірну мережу. Внутрішні маршрутизатори перенаправляють запити читання та запису через маршрути mesh, обираючи найкоротший шлях до контролера відповідного банку пам'яті. Латентність доступу всередині вузла визначається кількістю хопів у mesh, станом LLC-slice та швидкістю DRAM. Перевага такої архітектури полягає у відсутності вузлової конкуренції: кожен сокет отримує власний широкосмуговий інтерфейс до пам'яті, здатний обслуговувати одночасні звернення кількох ядер без глобального блокування.

Взаємодія між вузлами здійснюється через QPI (у поколіннях Haswell/Broadwell) або UPI (у Skylake-SP і новіших), які є високошвидкісними двонаправленими лінками з пакетоорієнтованою передачею даних і команд когерентності. UPI виконує не лише транспортування даних, а й повноцінно обслуговує когерентність між 10 сокетами: запити на читання, read-for-ownership, інвалідизаційні транзакції та підтвердження станів кешових ліній. На фізичному рівні UPI складається з кількох lane-груп, які працюють на високих частотах із внутрішнім кодом корекції помилок і механізмами retry. Це дає пропускну здатність від 10 до 30+ ГБ/с на лінк, однак ця величина є малою порівняно з локальною пропускною здатністю DRAM, яка може перевищувати 120–200 ГБ/с на сокет [2; 3].

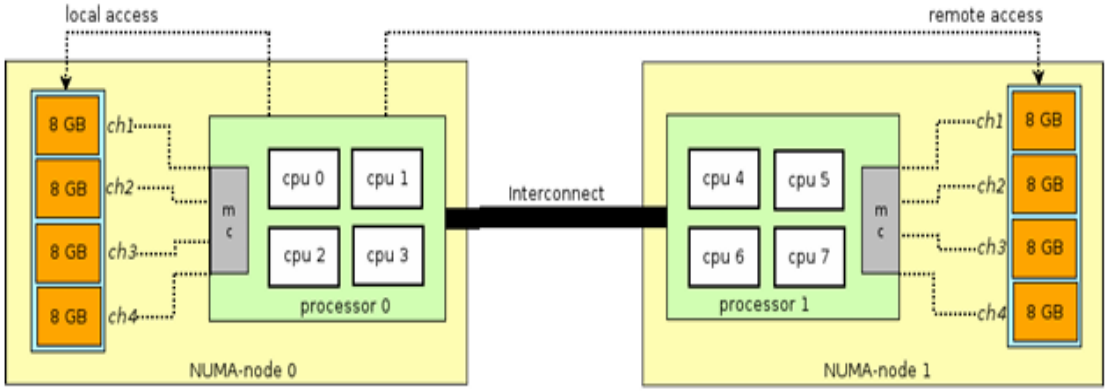


Рис. 1 Схема NUMA архітектури

Латентність доступу до локальної пам'яті у сучасних Xeon коливається в межах 70–100 нс, тоді як доступ до віддаленої пам'яті збільшує цей показник у два-три рази, часто до 180–250 нс і більше, залежно від топології UPI та навантаження когерентного трафіку. Причиною є не лише фізична відстань і швидкість лінка, а й когерентні процедури: кожен запит на читання повинен бути перевірений на наявність копій у кешах іншого сокета, а запит на запис повинен інвалідизувати всі потенційні копії лінії у віддалених кешах до того, як її можна буде модифікувати. Таким чином, NUMA формує фундаментальний принцип: продуктивність визначається не лише алгоритмом і тактовою частотою процесора, а й просторовою близькістю даних до потоку, який їх обробляє.

NUMA-топологія виявляється через SRAT та SLIT таблиці ACPI, які ядро операційної системи використовує для побудови карти відстаней між вузлами. У Linux значення у цій матриці відображають відносну латентність 11 доступів і впливають на рішення щодо прив'язки потоків, розподілу сторінок та евристик AutoNUMA. Якщо архітектура програми ігнорує фізичну топологію, система автоматично потрапляє під вплив remote access penalties, коли UPI стає вузьким місцем, когерентний трафік витісняє корисні дані, а кеш-підсистеми постійно займаються інвалідизацією, замість того щоб обслуговувати обчислення. Саме тому NUMA вимагає системного підходу: правильного розміщення даних, прив'язки потоків, виконання ініціалізації на тих вузлах, де відбуватиметься обробка, і мінімізації міжсокетних залежностей [4; 5]. Проте фізична наявність швидкісних лінків між вузлами є лише частиною рішення. Головна складність полягає у забезпеченні логічної цілісності даних, які одночасно обробляються різними сокетами. Це завдання покладається на складні апаратні протоколи, що керують станами кеш-ліній у масштабі всієї системи, роблячи **когерентність кешів у багатосокетних системах** критичним фактором продуктивності.

Протоколи когерентності MESIF та MOESI відповідають за логічну узгодженість копій кешових ліній, розташованих у кешах різних ядер і на різних сокетах. У багатосокетній системі когерентність стає критично важливою через те, що копії однієї й тієї ж cache line можуть одночасно існувати на різних вузлах, а будь-яке оновлення цієї лінії повинно бути глобально видимим. Когерентність не є абстрактним механізмом – кожен перехід у стани Modified, Exclusive, Shared, Invalid чи Owned/Foward вимагає фізичного трафіку через UPI, і саме цей трафік визначає реальну масштабованість паралельних алгоритмів.

У нормальному випадку кеш намагається читати дані локально. Якщо лінія відсутня в L1/L2/L3, ядро звертається до LLC slice, який виступає як directory-система. Якщо директорій показує, що лінія знаходиться на іншому сокеті в стані Modified або Owned, HA (Home Agent) ініціює міжсокетний snoop-запит. Цей запит передається через UPI і вимагає отримання відповіді від remote LLC, що додає мінімум десятки наносекунд до латентності вже після внутрішнього доступу. Якщо лінія модифікована, інший сокет повинен передати її дані, і лише після цього локальний сокет може кешувати її у стані Shared або Modified залежно від операції. Операції запису є значно дорожчими за читання, оскільки вони мають отримати ексклюзивне право на кешову лінію. Якщо будь-який інший сокет має копію лінії у стані Shared або Forward, HA ініціює інвалідизаційний цикл, який змушує інші кеші перевести лінію у стан Invalid. Весь цей цикл здійснюється через міжсокетні інтерконекти, що створює конкуренцію за UPI bandwidth. Якщо два ядра на різних сокетах записують у змінні, які лежать в одній лінії кешу, лінія переміщується між ними у стані Modified у циклі, що практично блокує виконання обох потоків через постійну синхронізацію. Це явище є найбільш руйнівною формою false sharing і становить одну з головних причин поганої масштабованості на багатосокетних машинах. У MESIF додатковий стан Forward дозволяє уникнути множинних відповідей на read-запит, але не вирішує проблему частих інвалідизацій. У MOESI стан Owned дозволяє одному ядру зберігати модифіковану копію лінії навіть тоді, коли інші читають її у стані Shared, але це також не усуває необхідність інвалідизацій під час записів. З погляду міжсокетного трафіку ці протоколи визначають, скільки повідомлень буде передано по UPI на кожен запис у спільні дані. На практиці синхронізація потоків, що працюють на різних вузлах, відбувається через кешові лінії: mutex – це змінна, що зберігається у кеші; atomic операції працюють через read-modify-write цикли над цією лінією; spinlock у постійній петлі читає одну і ту саму лінію, а CAS-запити генерують RFO, який щоразу переносить лінію у Modified стан. Це робить прямі блокування в NUMA-середовищах радикально дорожчими, ніж у односокетних системах, і пояснює чому lock-free структури, що побудовані з урахуванням NUMA, дають суттєву перевагу. Якщо кешова лінія змушена часто переміщуватися між сокетами, UPI стає вузьким місцем, а CPU на обох вузлах простоюють у очікуванні переходів між когерентними станами. Тому когерентність у NUMA-системах 13 – це не просто механізм узгодженості, а один із головних факторів, який визначає архітектуру паралельних програм. Проте сама по собі когерентність лише забезпечує коректність даних; справжня ж ефективність обчислень залежить від того, наскільки вдало дані територіально наближені до обчислювальних ядер. Саме тут на перший план виходять **політики NUMA-пам'яті**, які стають інструментом керування фізичним розміщенням даних у системі.

Політики NUMA-пам'яті визначають, на якому фізичному вузлі будуть розміщуватися сторінки процесу, і тим самим формують реальну топологію доступу до пам'яті для всіх потоків. Оскільки в системі з кількома сокетами кожна сторінка прив'язана до певного DRAM-банку, кожен доступ до неї змушений проходити або локальний шлях (для потоків на тому самому вузлі), або міжсокетний шлях через UPI (для потоків на інших вузлах). Політика розміщення визначає характер та інтенсивність віддалених доступів, які в нормальних умовах становлять основне джерело втрат продуктивності у NUMA-середовищах. Політика first-touch формує розподіл пам'яті виключно на основі того, який потік першим звертається до сторінки. Це означає, що архітектура

виконання програми у фазі ініціалізації визначає поведінку системи на усіх наступних етапах. Якщо більшість масивів ініціалізуються в одному робочому потоці, який scheduler розміщує на Node 0, ці сторінки фізично з'являються саме там. Коли обчислення розподіляються між потоками на різних вузлах, утворюється нерівномірний доступ, і вузол, який не володіє сторінками, змушений звертатися через UPI, створюючи remote load/store cycles. Внаслідок цього кожен cache miss перетворюється на багатоступеневий запит, що включає когерентність, маршрутизацію, доступ до remote DRAM і повернення даних назад. Політика interleave рівномірно розподіляє сторінки між усіма вузлами та створює симетричну структуру пам'яті. Її застосовують тоді, коли задача є bandwidth-bound, а не latency-bound. У такій конфігурації пропускна здатність пам'яті лінійно збільшується з кількістю вузлів, оскільки кожен вузол обслуговує свою частку доступів. Проте цей підхід має фундаментальну проблему: потік на будь-якому вузлі завжди виконуватиме частину доступів 14 до віддаленої пам'яті, тому його середня латентність ніколи не буде нижчою за середнє арифметичне між локальною та віддаленою. Таким чином, interleave – це завжди обмін локальної ефективності на глобальну рівномірність. Політика preferred забезпечує приоритезоване, але не жорстке розміщення на певному вузлі. Вона корисна у динамічних середовищах, у яких робочий набір потоків може змінюватись у runtime. У таких випадках ядро намагатиметься тримати сторінки на певному вузлі, але за потреби може переносити їх на інші. Це зрушує відповідальність з розробника на ОС, але викликає ризик непередбачуваності: сторінки, які вважалися локальними, можуть неочікувано виявитися розподіленими, що викликає складні схеми remote access. Найжорсткіша політика bind перетворює вузол на фізичний домен пам'яті, у якому програма може розміщувати свої структури без ризику розпорошення. Для NUMA-адаптованих алгоритмів bind дозволяє створити повністю локальний сегмент пам'яті, у якому залишаються всі робочі структури. Це дозволяє уникнути будь-яких віддалених доступів і створювати ізольовані обчислювальні зони. У складних системах, таких як високопродуктивні БД, розподілені key-value сховища, in-memory аналітика, такі вузли стають природними shard-ами даних, яких не торкається AutoNUMA або scheduler. Політики NUMA-пам'яті напряму взаємодіють із планувальником потоків. Якщо scheduler переміщує потік на інший вузол, first-touch може призвести до того, що потік тепер працює з повністю віддаленим робочим набором. Якщо AutoNUMA увімкнений, система може почати переміщувати сторінки у відповідь на це, що породить хвилю міграцій та TLB shootdowns, погіршуючи ситуацію. Саме тому поєднання NUMA-policy та thread affinity є обов'язковим елементом правильної архітектури багатопотокових програм.

Таким чином, ефективність NUMA-систем критично залежить від просторової локальності даних, оскільки доступ до віддаленої пам'яті через UPI збільшує затримки у кілька разів. Протоколи MESIF та MOESI забезпечують цілісність кешів, проте інтенсивний міжсокетний трафік та явища *false sharing* стають головними бар'єрами для масштабування. Мінімізація синхронізації між вузлами та використання *lock-free* структур є необхідною умовою для розкриття потенціалу багатосокетних платформ.

Вибір політики розміщення сторінок, як-от *first-touch* або *interleave*, дозволяє адаптувати систему під конкретний тип обчислювального навантаження. Найвищу продуктивність демонструє поєднання жорсткої прив'язки потоків (*thread affinity*) із політикою *bind*, що ізолює обчислення всередині локальних доменів. Такий підхід дозволяє уникнути непередбачуваних міграцій даних та накладних витрат на роботу планувальника й механізмів AutoNUMA.

Список літератури

1. Кіш В. В. Дослідження методів і програмних засобів оптимізації міжсокетної взаємодії у багатосокетних NUMA-системах : кваліфікаційна робота магістра : спец. 124 «Системний аналіз» / Кіш Віктор Вікторович. – Ужгород, 2025. – 80 с.

2. Intel Corporation. Intel® 64 and IA-32 Architectures Software Developer's Manual. 2024.

3. Jeffers, J., Reinders, J. Intel Xeon Phi Processor High Performance Programming. Morgan Kaufmann, 2016.

4. Molka, D., Hackenberg, D., Schöne, R. Memory Performance and NUMA Characteristics on Modern Multicore x86 Processors. IEEE IPDPS Workshops, 2015.

5. Wong, H., Papadopoulos, A., Tsafrir, D. The Impact of Kernel NUMA Balancing on Memory Locality. USENIX ATC, 2019.

УДК 004.5:004.88

Ю.В. Білявська
y.biliavska@knute.edu.ua
Державний торговельно-економічний університет, Київ

ПЕРЕВАГИ, ІНСТРУМЕНТИ ТА РИЗИКИ ЦИФРОВОГО ДВІЙНИКА

Цифрові технології стрімко проникли у життя суспільства, адаптували та змінили галузі промисловості, продовжують трансформувати бізнес-моделі підприємств й формувати подальшу стратегію їх подальшого розвитку. В роботі встановлено, що майбутнє пов'язано, як з розвитком цифрових технологій так і тими, хто їх активно впроваджує. Розвиток технологій Інтернету речей, Великих масивів даних, хмарних обчислень й штучного інтелекту призвів до формування певних можливостей у багатьох галузях [1]. Кількість підключених пристроїв до мережі Інтернет, формування ІТ екосистем щороку зростає, що допомагає згенерувати й опрацювати великий масив даних. Дедалі частіше набуває актуальності поєднання офлайн діяльності підприємств із онлайн складовою. Саме тому за допомогою цифрових двійників в режимі реального часу можна забезпечити повноцінний зв'язок між офлайн й онлайн секторами економіки. Вже доведено, що цифрові двійники – це віртуальні прототипи фізичного об'єкта або групи об'єктів, які призначені для моделювання їхньої поведінки. Віртуальні моделі можуть визначати стан фізичних об'єктів, а також прогнозувати й оцінити зміни. Цифрові технології відіграють вирішальну роль у підвищенні конкурентоспроможності економіки країни та у стимулюванні економічного зростання всього світового ринку [2]. Впровадження інноваційних цифрових технологій сприяє підвищенню конкурентоспроможності підприємств й продуктивності праці персоналу, зменшує витрати бізнесу, дозволяє зберігати й захищати конференційну інформацію, знижує перешкоди при виході підприємств на нові ринки збуту й має мультиплікативний ефект на розвиток економки країни загалом.

Цифрові двійники – це набір певних параметрів, які вичерпно описують фізичний об'єкт чи процес. Інколи це виглядає дуже наочно, тому що для зручності цифрові двійники візуалізуються на екрані. Виходить повноцінна цифрова копія реального предмета для роботи у спеціальному софті. Свого "двійника" може отримати окремий виріб, цілий технологічний процес, підприємство і навіть сектор промисловості. Ще недавно цифрове моделювання було статичним. Для точної симуляції процесів дані доводилося знімати та коригувати вручну. Сьогодні все змінилося: засоби IoT, відкриті API, штучний інтелект та засоби Big Data створюють «цифрових двійників» максимально чутливими оскільки модель може автоматично оновлюватися на основі даних, що постійно надходять. Це робить моделювання простішим та більш точним. Для ефективної побудови «цифрових двійників» слід враховувати переваги від їх застосування, а також інструменти й ризики, що можуть виникати в процесі їхньої реалізації (рис. 1).

Рис. 1. Переваги, інструменти та ризики формування цифрових двійників [3]

Технологія цифрового двійника стала дуже затребуваною, оскільки дозволяє вирішувати низку найскладніших завдань, а саме – допомагає створювати безпечне медичне обладнання та надійні машини. Цифрові двійники підприємства можуть змоделювати комплексне управління виробництвом та спланувати його у найдрібніших елементах.

Також, слід зазначити, що з розвитком штучного інтелекту з'явилася небезпека створення цифрових двійників – віртуальних копій реальних людей. Ця технологія несе низку загроз.

Проблеми з конфіденційністю. Для створення цифрових двійників достатньо декілька хвилин аудіо або відео контенту. Це ставить під загрозу конфіденційність, оскільки цифрова копія може імітувати зовнішній вигляд, голос і поведінку реальної людини [3].

Можливість маніпуляції та дезінформації. Цифрові двійники можуть використовуватися для поширення дезінформації або маніпуляції громадською думкою. Наприклад, цифрові двійники відомого лідера думок можуть робити заяви, що суперечать його справжнім переконанням.

Етичні питання. Створення та використання цифрових двійників порушує етичні питання. Хто несе відповідальність за дії, вчинені цифровими двійниками? Які права та обов'язки творців і «власників» цих віртуальних копій? Загроза безпеці, тобто «цифрові двійники» можуть використовуватися для здійснення кібератак. Зловмисники можуть отримати несанкціонований доступ і використати його для обману системи ідентифікації та доступу до конфіденційної інформації.

Для протидії загрозам, пов'язаним із впливом цифрових двійників, необхідно дотримуватися таких заходів, як: розробка нормативно-правової бази, що регулює створення й використання цифрових двійників; посилення механізмів захисту конфіденційності; створення систем для виявлення й блокування цифрових двійників, що використовуються для дезінформації або кібератак. Підвищення обізнаності про ризики, що пов'язані з цифровими двійниками, серед користувачів і фахівців. Отже, цифрові двійники – це потужний інструмент, який може змінити взаємодію з технологіями та комунікаціями. Однак, збалансований підхід щодо розвитку та регулювання цієї технології дасть змогу використовувати її потенціал, мінімізуючи ризики.

Складні цифрові моделі допомагають у технічно складному виробництві – для контролю виготовлення та прогнозованого обслуговування виробів. Тому, найбільш широко технології Digital Twins сьогодні застосовують у промисловості.

Оскільки концепція цифрових двійників на виробництві народилася в аерокосмічній галузі, найбільш широке застосування спочатку знайшла у гігантів індустрії. Boeing, Airbus, SpaceX – всі вони використовують «цифрових двійників» для оптимізації виробництва та прийняття ефективних управлінських рішень. Але й в інших галузях такі технології вкрай затребувані, активно використовуються в: автомобілебудуванні, електроніці та фармації.

Отже, у сучасному світі Web 3.0 цифрові двійники стають імперативом бізнесу, охоплюючи весь життєвий цикл активу й формуючи підґрунтя для цифровізації всіх процесів підприємства, його продуктів й послуг для можливості існування в умовах постійних змін. Сформовані чотири етапи розвитку цифрових двійників дозволили простежити шлях від концепції формування до впевненого застосування у різних галузях промисловості. Такі еволюційні зміни потребують детальнішої уваги щодо визначення: переваг, інструментів та можливих ризиків використання цифрових двійників, що узагальнено у моделі A.I.R. Дотримання визначеної моделі сприятиме покращенню можливості ухвалення рішень за рахунок моделювання різних сценаріїв застосування з мінімальними ризиками; оптимізації бізнес-процесів; підвищенню ефективності при вирішенні складних завдань.

Список літератури

1. Jiang, B., Cheng, T., Tsou, M. H., Zhu, D., & Ye, X. (2025). Advancing translational human dynamics research: bridging space, mind, and computational urban science in the era of GeoAI. Computational Urban Science, 5(1), 1-9.

2. Dembski, F., Wössner, U., Letzgus, M., Ruddat, M., & Yamu, C. (2020). Urban digital twins for smart cities and citizens: The case study of Herrenberg, Germany. Sustainability, 12 (6), 2307.

3. Biliavska, Y., & Biliavskyi, V. (2026). Digital Twins in the Context of Ensuring Sustainable Industrial Development. Acta Informatica Pragensia, 15(1), 198-220. doi: 10.18267/j.aip.291

УДК 004.4

Б.Ю. Вінтенко[1,2], Т.В. Смірнова[3], І.В. Миронець[2], О.А. Смірнов[3]
boris.vintenko@gmail.com, sm.tetyana@gmail.com, i.myronets@chdtu.edu.ua, dr.smirnovoa@gmail.com
[1] ПАТ "Науково-виробниче підприємство "Радій", Кропивницький, Україна
[2] Черкаський державний технологічний університет, Черкаси, Україна
[3] Центральноукраїнський національний технічний університет, Кропивницький, Україна

РОЗРОБКА МЕТОДУ ОЦІНКИ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ КОМП'ЮТЕРНО-ОРІЄНТОВАНИХ ПРОЦЕДУР СИСТЕМИ ПІДТРИМКИ ОПЕРАТОРІВ АЕС

Постановка проблеми. Сучасні атомні електростанції – це складні високотехнологічні об'єкти з розгалуженою структурою обладнання та цифрових систем керування. Процес управління енергоблоком здійснюється оперативною зміною з блочного щита за допомогою широкого арсеналу засобів візуалізації та контролю. Дії персоналу суворо регламентовані інструкціями, які сьогодні реалізуються як у традиційному паперовому, так і в цифровому (комп'ютерно-орієнтованому) форматі. Впровадження комп'ютерно-орієнтованих процедур дозволяє створювати інтелектуальні системи підтримки оператора, що працюють у режимі реального часу. Такі системи безперервно аналізують потік даних від суміжних систем енергоблока, автоматично ідентифікують умови для виконання тих чи інших операцій, чим суттєво знижують когнітивне навантаження на персонал і мінімізують ризик помилок. З огляду на критичну важливість таких систем, особлива увага приділяється їхній відмовостійкості. Використання методів самодіагностики, резервування та багатоверсійного програмування дозволяє вчасно виявляти збої, проте самого факту констатації відмови недостатньо. Для систем критичного застосування оператору необхідно розуміти ступінь деградації функцій системи. Це зумовлює потребу в розробці методів оцінки функціональної стійкості, які б надавали точну кількісну характеристику здатності системи виконувати свої завдання навіть за наявності часткових пошкоджень

Метою роботи є розробка методу чисельної оцінки функціональної стійкості елементів інформаційної системи підтримки оперативного персоналу АЕС.

Модель комп'ютерно-орієнтованих процедур системи підтримки оперативного персоналу на основі продукційних правил. У технологічних регламентах, які є основою для створення систем підтримки оперативного персоналу у даному дослідженні, вказані дії оператора, які мають бути виконані в залежності від вихідних умов та результатів попередніх дій. Елементи даних регламентів розділяються на два основні типи: правила (Rules) та дії (Actions). В термінології інтелектуальних систем дані елементи утворюють систему продукційних правил на основі дерев рішень. У рамках дослідження було розроблено модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу на основі графу. Вершинами графа є правила та дії, ребрами – переходи між діями згідно правил. Модель такої процедури формальна описана відповідною функцією. Спрацювання правил активує гілки дій, які має виконати оператор. Також правилами визначається факт виконання дії оператором. Для визначення факту спрацювання правила використовуються значення технологічних параметрів, визначені в реальному часі, та вагові коефіцієнти. В свою чергу, значення кожного технологічного параметру P визначається на основі набору сигналів, отриманих від суміжних систем, логікою та власними ваговими коефіцієнтами. Наприклад, при відключенні головного циркуляційного насоса, системою нормальної експлуатації формується дискретний сигнал #GCN01_OFF, що спричиняє відображення інформації про дану подію на табло блочного щита керування. Для комп'ютерно-орієнтованих процедур відключення насоса є параметром, а дискретний сигнал є таким, що формує даний параметр. Саме сигнали, отримані від суміжних систем, є визначальними даними для фіксації поточного стану правил процедури керування.

Метод визначення функціональної стійкості комп'ютерно-орієнтованих процедур. У межах даного дослідження розглядається здатність системи підтримки оператора функціонувати в умовах неповноти або недостовірності вхідних даних. Для забезпечення цієї властивості пропонується використання механізмів резервування різнорідних джерел інформації та розробка алгоритму адаптивного заміщення втрачених даних. Ключовим індикатором для активації такого заміщення обрано показник функціональної стійкості розподілених інформаційних систем. Під функціональною стійкістю системи підтримки оператора пропонується розуміти її здатність безперервно та в автоматичному режимі забезпечувати персонал актуальними кроками процедур керування, навіть за умови відмови окремих каналів зв'язку або виведення обладнання в ремонт. З огляду на ієрархічну структуру системи, комп'ютерно-орієнтовані процедури розглядаються як фрагмент розподіленої мережі, де вхідні сигнали надходять із зовнішніх інформаційних систем, на їхній основі обчислюються стани технологічних параметрів, на базі станів параметрів спрацьовують логічні правила, що формують фінальну інформацію для оператора. Такий підхід дозволяє розраховувати рівень функціональної стійкості на кожному рівні ієрархії, забезпечуючи надійну ідентифікацію стану енергоблоку за будь-яких умов. Дану ієрархію зобразимо у вигляді графу. Вершини {s} даного графу є

мажоритованими резервованими наборами сигналів, {P} – параметрами, {R} – правилами. Кожне ребро означає напрям передачі даних між елементами та має ваговий коефіцієнт ω. Джерелом негативних впливів на дану систему є можливі недостовірні або відсутні значення сигналів {s}, що з точки зору функціональної стійкості спричиняє розрив окремих ребер графу, тобто знижує вершинну зв'язність. Проте, вершинна зв'язність не є зручною в якості показника функціональної стійкості, оскільки кожний зв'язок між елементами має свою вагу (ω). Більш зручним є використання критерію ймовірнісної зв'язності розподіленої системи, що у теорії функціональної стійкості характеризує ймовірність встановлення зв'язку між парами вузлів. У контексті комп'ютерно-орієнтованих процедур даний критерій використаємо в якості показника ймовірності достовірного визначення стану її елементів.

Функціональна стійкість наборів сигналів. Сигнали, які є вхідною інформацією для комп'ютерно-орієнтованих процедур, об'єднані в набори з метою резервування. Кожний набір сигналів {s} містить інформацію від обраних з загальної множини сигналів, отриманих від зовнішніх джерел – інших програмно-технічних комплексів енергоблоку. Отримані вхідні сигнали проходять логічну обробку для формування технологічних параметрів за допомогою алгоритмів голосування, усереднення або визначення медіани. Дослідження базується на схемі резервування «M з N», де для верифікації параметра необхідна узгоджена робота щонайменше M сигналів із загальної кількості N. Кожному сигналу відповідає певна ймовірність безвідмовної роботи (наявності значення). Для кількісної оцінки стану системи введено такі рівні функціональної стійкості: Максимальний (100%): Сума ймовірностей безвідмовності всіх N сигналів, що відповідає повній структурній надмірності; Критичний: Сума мінімальних ймовірностей безвідмовності для гранично допустимої кількості сигналів M. Нижче цього рівня будь-яка одинична відмова призводить до повної непрацездатності набору; Поточний: Розраховується як ймовірність сукупності подій на основі фактично доступних у даний момент сигналів; Запас стійкості: Різниця між поточним і критичним показниками функціональної стійкості. Для зручності аналізу всі абсолютні значення конвертуються у відсотки за допомогою коефіцієнта пропорційності. Такий підхід дозволяє оперативно визначати функціональну стійкість технологічних параметрів, що лежать в основі комп'ютерно-орієнтованих процедур.

Функціональна стійкість правил комп'ютерно-орієнтованих процедур. Технологічні параметри використовуються для фіксації стану правил комп'ютерно-орієнтованих процедур {R} з використанням інформаційних вагових внесків {ω}. Таким чином, умовою можливості визначення стану параметрів та правил є не тільки наявність та узгодженість, а і достатня «вага» наборів даних. Для розрахунку функціональної стійкості правила кожний технологічний параметр P_i має мати попередньо розраховані показники функціональної стійкості. Під максимальним показником функціональної стійкості правила за середньою оцінкою будемо вважати середнє значення функціональної стійкості всіх технологічних параметрів, помножених на вагові коефіцієнти. За консервативною оцінкою за максимальний показник функціональної стійкості будемо вважати максимальне значення серед функціональної стійкості технологічних параметрів, помножених на вагові коефіцієнти. Це значення становить 100% функціональної стійкості правила. Під критичним та поточним показниками функціональної стійкості правила будемо вважати середні (для середньої оцінки) або мінімальні (для консервативної оцінки) значення серед відповідних рівнів функціональної стійкості всіх технологічних параметрів, помножених на вагові коефіцієнти: критичний середній; поточний середній; критичний консервативний; поточний консервативний. Запас функціональної стійкості визначається з різниці між поточною та критичною функціональною стійкістю. Усі визначені абсолютні значення функціональної стійкості можуть бути виражені у відсотках з використанням коефіцієнту.

Функціональна стійкість комп'ютерно-орієнтованих процедур та систем підтримки оперативного персоналу. На основі обчислених показників функціональної стійкості кожного правила може бути обчислена функціональна стійкість всієї комп'ютерно-орієнтованої процедури. В свою чергу, обчислення функціональної стійкості всіх комп'ютерно-орієнтованих процедур надає можливість оцінити функціональну стійкість всієї системи підтримки оперативного персоналу. Виходячи з того, що у даному дослідженні розглядається створення систем підтримки оперативного персоналу для АЕС як системи критичного застосування, для визначення функціональної стійкості комп'ютерно-орієнтованих процедур та систем підтримки оперативного персоналу пропонується використовувати консервативну оцінку, яка являє собою найнижче поточне значення функціональної стійкості серед усіх правил.

Реалізація оцінки функціональної стійкості комп'ютерно-орієнтованих процедур та моделювання відмов. До складу розроблюваного прототипу системи підтримки оперативного персоналу входить програма OpSupport, що виконує функції власне інформаційної підтримки оператора. Вона оперує наступними даними: значення вхідних сигналів, що отримуються мережевим протоколом від серверів програмно-технічних комплексів енергоблоку; базу даних технологічних параметрів, для кожного з яких визначені кластеризовані сигнали та скрипти розрахунку; базу даних правил та дій. Під час функціонування програми проводиться безперервний розрахунок показників функціональної стійкості для технологічних параметрів, правил та дій. Програма виконує розрахунок функціональної стійкості на основі показників прогнозованої ймовірності надійної роботи сигналів та визначеної логіки мажоритування M/N. Розрахунок обчислює максимальний, поточний та критичний рівні функціональної стійкості у відсотках. Критичний рівень розраховується як сума ймовірностей роботи M сигналів, відсортованих за зростанням (тобто мінімальним показником ймовірності визначення, при якому можлива робота). Поточний визначається як сума ймовірностей визначення значення N

сигналів, що знаходяться в роботі в даний момент. Для дослідження ефективності даних алгоритмів була змодельована база знань систем підтримки оперативного персоналу з об'ємом, що відповідає документу інструкції з ліквідації порушень: 50 процедур, до 10 правил активації процедури, до 3-х правил завершення процедури, до 10 кроків на процедуру, до 10 підкроків на кожний крок процедури, до 3 умов активації та завершення кроку. В якості вхідних даних було змодельовано 1000 сигналів, ймовірність безвідмовної роботи кожного з яких випадковим чином становить від 0.95 до 1. Для проведення експериментів було реалізовано можливість імітування відмов заданої кількості сигналів. В кожному експерименті була імітована відмова 20 випадкових сигналів та збережено результати зниження функціональної стійкості. Кожний експеримент проводився 10 разів за однакових умов. Середнє значення критичного рівня функціональної стійкості технологічних параметрів, що відмовили, становило 66.23%. Середнє значення поточного рівня функціональної стійкості становило 97.96%. Відповідно, запас функціональної стійкості у середньому становив 31.03%. При розрахунку функціональної стійкості правил, дій та процедур реалізована можливість використання двох оцінок: середньої та консервативної. Середня оцінка функціональної стійкості обчислюється як середнє арифметичне відповідного показника функціональної стійкості дочірніх елементів моделі комп'ютерно-орієнтованих процедур, тобто дочірніх правил та дій. Консервативна оцінка обирається за принципом мінімальної («найгіршої») оцінки серед усіх наявних. Середня оцінка є більш «чутливою» і надає можливість точніше оцінювати вплив відмов сигналів на працездатність елементів комп'ютерно-орієнтованих процедур. Проте у системі критичного застосування, коли система або її частина вважається повністю непрацездатною при повній відмові хоча б одного елемента, більш вірним є використання консервативної оцінки для фіксації відмов. Під час кожного експерименту також було зафіксовано зниження поточного функціональної стійкості систем підтримки оперативного персоналу за середньою та консервативною оцінкою. При консервативній оцінці критичний рівень функціональної стійкості системи становив 65.78%, поточний у середньому 74.35%. При цьому середній запас функціональної стійкості становив 8.57%. При середній оцінці критичний рівень функціональної стійкості становив 75.6%, поточний – 98.8%. При цьому середній запас функціональної стійкості становив у середньому 23.2%.

Висновки. У роботі впроваджено концепцію функціональної стійкості як інструменту оперативного оцінювання відмовостійкості компонентів комп'ютерно-орієнтованих процедур. Запропоновано методику розрахунку трьох ключових рівнів стійкості: максимального, поточного та критичного. Практичну придатність підходу підтверджено на прикладі алгоритмів інструкції з ліквідації порушень на АЕС. Моделювання відмов вхідних сигналів виявило суттєве скорочення запасу стійкості за консервативним сценарієм. Подальші дослідження будуть спрямовані на розробку методів підвищення надійності систем підтримки персоналу, ефективність яких визначатиметься за приростом показника запасу функціональної стійкості

Список літератури

1. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». Кібербезпека: освіта, наука, техніка. 2025. Том 2 № 30. С. 413-427, 2025.

2. Барабаш О.В. Побудова функціонально стійких розподілених інформаційних систем / О.В. Барабаш. – К.: НАОУ, 2004. – 226 с.

3. Інструкція з ліквідації порушень нормальної експлуатації енергоблоку №4 РАЕС. ВП «Рівненська АЕС», 2024. 286 с.

4. Вінтенко, Б., Миронець, І., Смірнов, О., Коваленко, А., Конопліцька-Слободенюк , О., Смірнова , Т., Константинова, Л. «Дослідження застосування систем підтримки оперативного персоналу об'єкту критичної інфраструктури при керуванні енергоблоком АЕС з реактором типу ВВЕР-1000». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 6-26.

5. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуй А.М. «Модель шляхів отримання вхідних даних комп'ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». *Центральноукраїнський науковий вісник. Технічні науки.* 2025. Вип. 11(42), ч. II. С.52-62.

УДК 004.8:004.4

Н.Л. Козірова, Р.О. Ткачук, П.С. Усік, Г.М. Дреєва
natalidonchenko23@gmail.com, tkachukroman64@gmail.com, usikps@kntu.kr.ua, gannadreeva@gmail.com
Центральноукраїнський національний технічний університет. Кропивницький

ПАРАДИГМА AGENTIC SDLC: ТРАНСФОРМАЦІЯ РОЛІ ІНЖЕНЕРА У ПРОЦЕСІ АВТОНОМНОЇ РОЗРОБКИ ПЗ

У роботі досліджується еволюція життєвого циклу розробки програмного забезпечення (SDLC) під впливом інтелектуальних агентів. Автором запропоновано трирівневу модель інтеграції AI, що описує перехід від допоміжних інструментів до автономних екосистем. Проаналізовано зміну когнітивного навантаження програміста та обґрунтовано концепцію Agentic SDLC, де людина виконує роль стратегічного координатора та верифікатора.Сучасна програмна інженерія стикається з кризою складності: обсяги кодових баз зростають швидше, ніж здатність людини їх підтримувати. Особливого значення проблема набуває в аспекті вільного програмного забезпечення (OpenSource), де більша частина коду підтримується волонтерами та більшість цікавих та важливих проєктів зникають через нестачу розробників. Інтеграція Large Language Models (LLM) та спеціалізованих агентних архітектур (наприклад, Single-Threaded Master Loop) дозволяє автоматизувати не лише написання коду, а й прийняття інженерних рішень[1].

Рівні інтеграції та автономності AI

Для глибшого розуміння процесів трансформації пропонується класифікація за ступенем когнітивної автономності систем:

Level 1: AI-Augmented Development (Асистивна розробка)

На цьому рівні AI функціонує як контекстне доповнення. Основна технологія — Neural Code Completion. Агенти працюють у межах одного файлу або функції, пропонуючи варіанти завершення рядків або генерацію типових "boilerplate" конструкцій.

- Глибина процесу: Людина зберігає повний контроль над логікою. AI лише прискорює "typing speed" та зменшує когнітивне навантаження на запам'ятовування синтаксису бібліотек.

- Ключові інструменти: GitHub Copilot, Tabnine, AWS CodeWhisperer.

- Результат: Підвищення індивідуальної продуктивності розробника на 20-50% у рутинних задачах.

- Переваги: Мінімальний поріг входу; значне скорочення часу на написання типового коду; допомога у вивченні нових синтаксичних конструкцій.

- Недоліки: Ризик "сліпого копіювання"; можливість виникнення дрібних синтаксичних помилок; відсутність розуміння загального архітектурного контексту проєкту[1].

Level 2: Task-Oriented Autonomous Agents (Агенти виконання задач)

Це перехід до систем, що володіють здатністю до Reasoning & Planning (логічного виведення та планування). Агент отримує не фрагмент коду, а цілісну інженерну задачу (наприклад, "створити API-ендпоінт для авторизації").

- Глибина процесу: Агент самостійно аналізує файлову структуру, обирає необхідні залежності, створює декілька файлів та запускає базові тести для самоперевірки за паттерном Self-Reflection. Людина взаємодіє на рівні перевірки Pull Request.

- Ключові інструменти: Devin, OpenDevin (Codium), Sweep.dev.

- Результат: Делегування цілісних функціональних модулів під наглядом інженера.

- Переваги: Делегування цілих робочих потоків; автоматизація рутинного тестування та документації; можливість паралельного виконання декількох інженерних задач.

- Недоліки: Висока вартість обчислень (токенів); складність контролю логіки в об'ємних PR; ризик внесення латентних логічних помилок, які важко помітити при рев'ю[2,4].

Level 3: Agentic SDLC & Multi-Agent Orchestration (Агентна екосистема)

Найвищий рівень, де AI інтегрований у всі етапи життєвого циклу як мережа взаємодіючих сутностей (Multi-Agent Systems - MAS). Кожен агент має свою роль: "Product Manager Agent" генерує вимоги, "Architect Agent" проєктує структуру, "Developer Agent" пише код, а "QA Agent" автономно шукає вразливості.

- Глибина процесу: Це створення автономного циклу розробки. Системи здатні до Self-healing (самовідновлення): якщо моніторинг фіксує помилку на продуктивному сервері, агент автономно створює фікс, тестує його та пропонує до деплою.

- Ключові інструменти: Claude Code, Microsoft AutoGen, LangGraph.

- Результат: Формування "AI-заводу" ПЗ, де роль людини — це стратегічний контроль та фінальна валідація (Sign-off).

- Переваги: Максимальний Time-to-Market; безперервна оптимізація системи 24/7; радикальне зниження навантаження на людський ресурс у супроводі складних систем.

- Недоліки: Knowledge Erosion (втрата фундаментальних навичок людьми); проблема "чорної скриньки" (складність інтерпретації архітектурних рішень AI); ризик неконтрольованого накопичення специфічного технічного боргу. В перспективі втрата великої кількості професійних кадрів[5,6].

Таблиця 1

Параметри оцінки ефективності на різних рівнях моделі Agentic SDLC

Параметр	Level 1 (Assistive)	Level 2 (Task-Oriented)	Level 3 (Agentic SDLC)
Одиниця роботи	Рядок / Функція	Задача (Issue) / Модуль	Продукт / Екосистема
Роль людини	Виконавець (Coder)	Менеджер (Task Lead)	Стратег (Director)
Ефективність	Висока для простих задач	Середня (потребує ітерацій)	Залежить від архітектури
MTTR (Час ремонту)	Знижується помірно	Знижується суттєво	Мінімальний (автономний)

Оцінка ефективності впровадження Agentic SDLC[1,6]

Для об'єктивного аналізу успішності інтеграції AI-агентів у робочі процеси пропонується використовувати систему метрик, що базується на швидкості, автономності та якості рішень:

- Pass@k (Показник успішності генерації): Кількісна метрика якості коду. Якщо ми просимо AI запропонувати k варіантів розв'язання задачі ($k=1, 5, 10$), показник вважається позитивним, якщо хоча б один із цих варіантів успішно проходить функціональні тести. Це дозволяє нівелювати стохастичну (випадкову) природу нейромереж.

- Resolution Rate (Коефіцієнт автономного вирішення): Відсоток інженерних завдань (наприклад, GitHub Issues або Jira Tickets), які агент зміг виконати повністю самостійно — від аналізу коду до створення Pull Request, який був прийнятий без суттєвих правок людиною.

- Human-to-Agent Effort Ratio (Співвідношення зусиль): Метрика, що показує, скільки часу витрачає інженер на нагляд та перевірку результатів роботи AI у порівнянні з часом, який агент витрачає на автономну роботу. Цей показник демонструє реальну економічну вигоду від автоматизації.

- Time-to-Market (Час виходу на ринок): Сумарний період від виникнення ідеї або формування технічного завдання до моменту доступності функціоналу для кінцевого користувача. В Agentic SDLC цей показник скорочується завдяки паралелізації роботи декількох агентів.

- MTTR (Mean Time to Repair — Середній час відновлення): Швидкість, з якою AI-агенти виявляють, локалізують та виправляють критичні помилки або вразливості в системі без прямого втручання розробника (Self-healing).

Ці метрики дозволяють не просто сказати "AI працює добре", а довести це цифрами, що є обов'язковим для серйозної наукової чи аналітичної роботи.

Висновки

Впровадження Agentic SDLC змінює когнітивну структуру програмної діяльності. Програміст майбутнього має змістити фокус із написання синтаксису на System Design та Security Audit. Основним викликом стає не генерація коду, а забезпечення його цілісності та верифікація рішень, прийнятих автономними агентами. Подальші дослідження мають бути спрямовані на розробку формальних моделей взаємодії "людина-AI" та методів автоматизованого контролю якості згенерованих архітектур.

Список літератури

1. Evaluating large language models trained on code / M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan et al. 2021. 42 p. URL: https://arxiv.org/abs/2107.03374

2.ReAct: synergizing reasoning and acting in language models / S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao // Proceedings of the 11th International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda, May 1–5, 2023. San Diego : ICLR, 2023. P. 1–33.

3. Tree of thoughts: deliberate problem solving with large language models / S. Yao, D. Yu, J. Zhao, I. Shafran, K. Narasimhan, Y. Yuan, Y. Cao. 2023. 15 p. URL: https://arxiv.org/abs/2305.10601

4. Shinn N., Labash B., Gopinath A. Reflexion: language agents with iterative design learning. 2023. 32 p. URL: https://arxiv.org/abs/2303.11366

5.AutoGen: enabling next-gen LLM applications via multi-agent conversation / Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu et al. *Microsoft Research Technical Report*. Redmond : Microsoft, 2023. No. MSR-TR-2023-35. 25 p.

6.Hummer W., Rosenberg F. The evolution of SDLC: integrating autonomous agents into the software supply chain. *Journal of Software Engineering & Applications*. 2024. Vol. 17, No. 2. P. 45–62.

УДК 519.85

В.І. Петренюк
petrenjukvi@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

ВИКОРИСТАННЯ МІНОРІВ ПРОЕКТИВНОЇ ПЛОЩИНИ ДЛЯ СИНТЕЗУ МОДЕЛЕЙ ГРАФІВ-ОБСТРУКЦІЙ ПОВЕРХНІ КЛЕЙНА

Задачу побудови моделей графів-обструкцій та мінорів поверхні Клейна шляхом синтезу пари мінорів проективної площини, які мають частинні чи породжені підграфи гомеоморфні $K_{3,3}$, або такі що стягуються до них при стисканні в точку, принаймні, одного ребра.

На думку автора задача є актуальною. Використано методи направленого синтезу та φ-перетворення пар мінорів проективної площини з підграфом гомеоморфним одному й тому графу Куратовського. Направлений синтез оперує попарно сумісними мінорами. Фі-перетворення пари мінорів проективної площини полягатиме у ототожнення пар відповідних вершин підграфів гомеоморфних, наприклад $K_{3,3}$.Сумісність це існування розширення мінімального вкладення першого графа впорядкованої пари в поверхню Клейна до вкладення другого мінора пари в одну з кліток поверхні Клейна з приклеєною до цієї клітки лентою Мебіуса.

Для побудови моделей мінорів поверхні Клейна потрібно спочатку дослідити структуру проективних підграфів $G \setminus v$, де v - задана вершина інцидентна вершинам множини M мінора G проективної площини. Також треба побудувати множину всіх неізоморфних вкладень підграфа $G \setminus v$, що розташовують кожну точку w з множини M на границі однієї клітки, а решту точок з множини $M \setminus w$ на границі другої клітки. Потім треба побудувати для цих вкладень найкоротші кліткові ланцюги з двох кліток із, принаймні, однією спільною граничною точко, будуть мати кліткову довжину 1, які з'єднують граничні цикли із точками заданої множини M точок графа $G \setminus v$.

Перелік графів, як задіяних у синтезу, буде наведено на репозитарії університету. Всі різні, з точністю до ізоморфізму, зв'язні підграфи $G \setminus v$ мінорів G проективної площини з породженим підграфом $K_{3,3}$ в додатку А, де червоними є точки з множини M відповідної вершини v. Додаток 1 має 109 2- та 3-зв'язних моделей мінорів поверхні Клейна синтезованих як φ-образи $C_3 \setminus 1$ та допустимих мінорів проективної площини. Додаток 2 містить список з 247 моделей мінорів поверхні Клейна синтезованих по $K_{3,3}$ з пари $(F_6 \setminus 2, G)$, де G-довільний 2-зв'язний мінор проективної площини йде першим, потім його підграфи виду $G \setminus v$. Для побудови 3-зв'язних мінорів шляхом рекурсивного φ-перетворення 2-зв'язного мінора та простої зірки шляхом ототодженні пар точок 2-зв'язного з клітковою відстанню 1 та висячих вершин зірки.. В результаті доведено теорему, де досліджено синтез графів із наперед заданим неорієнтованим родом утвореного графа, з упорядкованої пари мінорів проективної площини. Графи-обструкції чи мінори поверхні Клейна можливо знайти через φ-перетворення пар -мінорів проективної площини шляхом ототожнення відповідних пар вершин породженого чи частинного підграфа $K_{3,3}$ та в результаті операцій видалення чи стискання в точку ребер, що є несуттєвими відносно роду 3. Наведено алгоритм побудови моделей мінорів поверхні Клейна, яким отримано неповний список цих графів, наведений в додатках 1 та 2..

Список літератури

1. Хоменко М. П. φ-перетворення графів. Препринт ИМ АНУ. Киев. 1973. 383 с.
2. Хоменко М. П. Топологические аспекты теории графов. Препринт ИМ АНУ. Киев. 1970. 299 с.
3. Mohar B., Thomassen C. Graphs on Surfaces. Johns Hopkins University Press, 2001. 412 p. https://www.sfu.ca/~mohar/Book.html
4. Hur S. The Kuratowski covering conjecture for graphs of order less than 10. Phd, Ohio State University, 2008. http://rave.ohiolink.edu/etdc/view?acc_num=osu1209141894
5. Archdeacon D., Huneke P. A Kuratowski Theorem for Nonorientable Surfaces. Journal of combinatorial theory, Series B. 1989. **46**. P. 173–231.
6. Петренюк В.І. Про структуру площинних підграфів графів-обструкцій неорієнтованої поверхні заданого роду. Фізико математичне моделювання та інформаційні технології. 2021. № 33. С. 105–109. Google Scholar.
7. Anna Flötotto. Embeddability of graphs into the Klein surface. Dissertation, University Bielefeld, 2010, -174 pp.
8. P.Skoda. Obstructions for embedding graphs into surfaces**,** Simon Frazer University, PhD dissertation, 2012.-133 p.

УДК 004.4

В.П. Кулагін, *аспірант, 2 курс*
ПВНЗ «Європейський університет», Київ
Наук. керівник. О.С. Улічев, *канд. техн. наук*
victor@kulagin.com.ua, askin79@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

РОЗВИТОК ІГРОВИХ МОДЕЛЕЙ ДЛЯ ОПТИМІЗАЦІЇ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ: ВІД РІВНОВАГИ НЕША ДО КООПЕРАЦІЙ ТА БАГАТОРІВНЕВИХ ВЗАЄМОДІЙ

Мікросервісна архітектура посідає важливе місце у розробці сучасних програмних систем завдяки гнучкості, масштабованості та незалежного розгортання компонентів [1-3]. Водночас перехід від монолітних рішень до множини автономних сервісів породжує новий клас проблем, пов'язаних із динамічним розподілом обчислювальних ресурсів, забезпеченням QoS, дотриманням SLA та узгодженням локальних цілей окремих сервісів із глобальною ефективністю системи [1].

У статті «Оптимізація роботи мікросервісів на основі теорії ігор» запропоновано підхід, у межах якого кожний мікросервіс розглядається як раціональний гравець, що прагне максимізувати власну корисність при спільному використанні обмежених ресурсів. Автори формалізують задачу як некооперативну гру, використовують концепцію рівноваги Неша та вводять Game Theory Controller, який на основі поточних метрик навантаження обчислює новий розподіл ресурсів між сервісами. Результати моделювання, наведені у статті, показують перевагу такого підходу порівняно з евристичними стратегіями Binpack і Spread за критерієм середнього часу відповіді системи [1].

Запропонована модель є важливим кроком у напрямі інтелектуального керування мікросервісними середовищами, однак самі автори вказують на обмеження роботи: аналіз виконується для обмеженої кількості сервісів, взаємозв'язки між сервісами враховано спрощено, а як перспективу подальших досліджень окреслено перехід до багаторівневих ігор та ігор з коаліціями [1]. Саме цей напрям видається найбільш перспективним, оскільки реальні мікросервісні системи рідко є набором повністю незалежних агентів. На практиці сервіси утворюють ланцюжки викликів, спільно реалізують бізнес-процеси, мають асиметричну критичність і нерівномірно впливають на загальну затримку та відмовостійкість системи.

Класична некооперативна модель з індивідуальною функцією виграшу добре описує конкуренцію за CPU, пам'ять або мережеву пропускну здатність, але недостатньо точно відображає колективну поведінку сервісів у межах одного бізнес-сценарію. Наприклад, якщо група сервісів спільно обслуговує критичний процес оплати, оформлення замовлення або аутентифікації, то доцільно розглядати її не лише як сукупність окремих гравців, а і як коаліцію з частково спільною цільовою функцією. У такому випадку оптимізація має враховувати не тільки локальну затримку окремого мікросервісу, але й сумарний внесок коаліції в наскрізний час виконання запиту, стабільність бізнес-транзакції та ризик деградації залежних компонентів [1, 6].

Теоретичне підґрунтя для такого переходу існує в межах теорії потенційних ігор, загальної рівноваги Неша та кооперативних моделей. Ще класична праця Дж. Неша заклала основу для аналізу стратегічної поведінки раціональних агентів [4], а робота D. Monderer і L. S. Shapley показала, що потенційні ігри мають важливі властивості збіжності та придатні для побудови розподілених механізмів узгодження рішень [5]. Для хмарних систем і сервісних платформ ідеї узагальненої рівноваги Неша вже застосовувались до задач сервісного забезпечення та розподілу ресурсів між кількома учасниками [7], що створює добру основу для перенесення цих підходів у мікросервісне середовище.

Особливо показовою є праця R. Luo та співавторів, у якій runtime-керування ресурсами для мікросервісних застосунків моделюється як congestion game [6]. У такій постановці сервіс прагне мінімізувати час відповіді та штрафи за порушення SLA, а перевантаження спільного ресурсу безпосередньо зменшує індивідуальний виграш. Саме цей підхід демонструє, що теорія ігор може бути не лише інструментом абстрактного аналізу, а й основою практичного контролера для оркестрації ресурсів. Проте congestion game здебільшого описує конкурентну взаємодію, тоді як у реальному мікросервісному середовищі присутні і кооперація, і ієрархічне підпорядкування, і адаптація до змінних умов навантаження [1, 6].

У цьому контексті доцільно запропонувати розвиток базової моделі [1] у трьох напрямах.

Перший напрям - багаторівнева гра. На верхньому рівні можуть діяти бізнес-домени або групи мікросервісів, які конкурують за частки кластера чи бюджет ресурсів. На нижньому рівні окремі сервіси всередині домену узгоджують локальний розподіл CPU, пам'яті, кількості реплік і пріоритетів обробки запитів. Така декомпозиція дозволить поєднати глобальну оптимізацію на рівні системи з локальною адаптацією на рівні сервісів. Подібна постановка краще відповідає реальній структурі великих платформ, де рішення приймаються не в одній точці, а на кількох рівнях абстракції [1, 7].

Другий напрям - ігри з коаліціями. У мікросервісній системі природними коаліціями можуть виступати сервіси одного bounded context, сервіси одного критичного бізнес-потоку або компоненти, пов'язані жорсткими

залежностями у графі викликів. У кооперативній постановці можна оцінювати не лише індивідуальний виграш сервісу, а й колективний виграш коаліції, а також справедливий розподіл ресурсів усередині неї. Це особливо важливо для сценаріїв, де деградація одного другорядного сервісу може бути прийнятною, якщо вона дозволяє зберегти стабільність для критичних сервісів того самого бізнес-потоку. Схожі ідеї кооперативної оптимізації вже використовувалися в edge/fog-середовищах, де потрібно одночасно балансувати енергоефективність, доступність і продуктивність [1, 8].

Третій напрям - еволюційні та адаптивні ігри. На відміну від статичної постановки, у реальних системах навантаження, топологія викликів і стан інфраструктури безперервно змінюються. Тому стратегія сервісу також має еволюціонувати на основі повторюваних взаємодій, історії виграшів і спостережуваного стану середовища. Еволюційні моделі вже показали ефективність у задачах вибору режимів доступу та сервісної селекції у fog/IoT-системах [1, 9], а отже можуть бути адаптовані і для мікросервісних платформ, де необхідно забезпечити стійке наближення до ефективного стану без жорстко централізованого керування.

Наукова новизна такого підходу полягає в переході від спрощеної моделі «один сервіс - один гравець» до моделі, де враховуються топологія викликів, критичність сервісів, колективні цілі бізнес-потоків та багаторівневий характер прийняття рішень. Практична цінність полягає у можливості побудови нового покоління контролерів оркестрації, які не лише реагують на перевищення порогів CPU або пам'яті, але й приймають рішення з урахуванням структурної ролі сервісу в системі. Це може покращити дотримання SLA, зменшити середній та хвостовий час відповіді, а також підвищити стійкість системи до пікових навантажень [1, 6, 7].

Список літератури

1. Улічев О. С., Кулагін В. П. Оптимізація роботи мікросервісів на основі теорії ігор. Центральноукраїнський науковий вісник. Технічні науки. 2025. Вип. 12(43), ч. 1. С. 44–57. DOI: 10.32515/2664-262X.2025.12(43).1.44-57.

2. Кулагін В. П., Улічев О. С., Доренський О. П. Інноваційні рішення та переваги мікросервісної архітектури програмних продуктів. Центральноукраїнський науковий вісник. Технічні науки. 2024. Вип. 10(41), ч. 1. С. 16–29. DOI: 10.32515/2664-262X.2024.10(41).16-29.

3. Dragoni N., Lanese I., Larsen S. T., Mazzara M., Mustafin R., Safina L. Microservices: How to make your application scale. Lecture Notes in Computer Science. 2018. Vol. 10742. P. 95–104. DOI: 10.1007/978-3-319-74313-4_8.

4. Nash J. F. Equilibrium points in n-person games. Proceedings of the National Academy of Sciences. 1950. Vol. 36, № 1. P. 48–49. DOI: 10.1073/pnas.36.1.48.

5. Monderer D., Shapley L. S. Potential Games. Games and Economic Behavior. 1996. Vol. 14, № 1. P. 124–143. DOI: 10.1006/game.1996.0044.

6. Luo R., Ye W., Sun J., Liu X., Zhang S. Runtime Resource Management for Microservices-Based Applications: A Congestion Game Approach. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. 2019. Vol. 268. P. 676–687. DOI: 10.1007/978-3-030-12981-1_47.

7. Ardagna D., Panicucci B., Passacantando M. Generalized Nash equilibria for the service provisioning problem in cloud systems. IEEE Transactions on Services Computing. 2013. Vol. 6, № 4. P. 429–442. DOI: 10.1109/TSC.2012.14.

8. Kaur K., Dhand T., Kumar N., Zeadally S. Container-as-a-Service at the Edge: Trade-off between Energy Efficiency and Service Availability at Fog Nano Data Centers. IEEE Wireless Communications. 2017. Vol. 24, № 3. P. 48–56. DOI: 10.1109/MWC.2017.1600427.

9. Yan S., Peng M., Abana M. A., Wang W. An Evolutionary Game for User Access Mode Selection in Fog Radio Access Networks. IEEE Access. 2017. Vol. 5. P. 2200–2210. DOI: 10.1109/ACCESS.2017.2654266.

УДК 004.45:004.056

О.К. Конопліцька-Слободенюк , А. О. Федотов, А.С. Коваленко
andrejfedotov012@gmail.com, ksuha80@gamail.com
Центральноукраїнський національний технічний університет, Кропивницький

ОГЛЯДСУЧАСНИХ ПАРАДИГМ ПРОГРАМУВАННЯ, ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТА АРХІТЕКТУРНИХ ПАТЕРНІВ В МОБІЛЬНІЙ ІНЖЕНЕРІЇ

У 2026 році глобальна екосистема розробки мобільних додатків перебуває на етапі фундаментальних трансформацій, які докорінно змінюють інженерні підходи до створення програмного забезпечення. З огляду на те, що кількість користувачів мобільних пристроїв у світі стрімко зростає, а сукупний обсяг ринку мобільних додатків впевнено долає нові економічні рубежі, мобільні платформи остаточно закріпили за собою статус домінуючого середовища для взаємодії користувача з цифровим світом. Програмування для мобільних платформ еволюціонувало від простого перенесення веб-інтерфейсів на екрани смартфонів до створення надскладних, автономних та криптографічно захищених екосистем.

Сьогодні інженери-програмісти стикаються з новими, безпрецедентними викликами: необхідністю забезпечити високу продуктивність при обмежених апаратних ресурсах, підтримкою різноманітних форм-факторів (включаючи складані пристрої з кількома екранами) та інтеграцією зі складною хмарною інфраструктурою. Саме архітектурні рішення, оптимізація коду та вибір оптимального технологічного стеку стають центральним об'єктом дискусій на провідних наукових майданчиках. Ця робота присвячена глибокому технічному огляду сучасних парадигм програмування, інструментальних засобів та архітектурних патернів, що домінують у мобільній інженерії поточного року.

Протягом останнього десятиліття архітектурні парадигми визначалися постійним вдосконаленням нативних засобів розробки. Нативна розробка залишається критично важливою для додатків, що вимагають максимальної продуктивності, плавності інтерфейсу (60-120 fps) та глибокої, низькорівневої інтеграції з апаратним забезпеченням і найновішими API операційних систем (iOS та Android).

Сучасні нативні переваги у 2026 році значною мірою базуються на масовому переході до декларативних інтерфейсів користувача (UI) та реактивного програмування. Інструменти SwiftUI для екосистеми Apple (мова Swift) та Jetpack Compose для Android (мова Kotlin) докорінно змінили процес побудови візуальної частини додатків. Декларативний підхід дозволяє програмістам просто описувати, як повинен виглядати інтерфейс при певному стані даних, залишаючи системі завдання самостійно обчислювати переходи та оновлення екрану під капотом. Це суттєво зменшує кількість шаблонного коду (boilerplate code), знижує ймовірність помилок, пов'язаних з розсинхронізацією стану, та робить код більш читабельним і передбачуваним.

Одночасно з цим еволюціонували й архітектурні патерни. Індустрія масово відійшла від застарілих шаблонів MVC (Model-View-Controller) та MVP (Model-View-Presenter), які часто призводили до створення так званих «масивних контролерів» (Massive View Controllers), що ускладнювали підтримку та тестування коду. У 2026 році стандартом де-факто для мобільної розробки стали патерни MVVM (Model-View-ViewModel) та MVI (Model-View-Intent). Ці підходи забезпечують чіткий однонаправлений потік даних (Unidirectional Data Flow), де стан додатку є незмінним (immutable), а будь-які зміни відбуваються виключно через інтенції (intents) або події, що генеруються користувачем. Такий архітектурний дизайн робить мобільні додатки високотестованими та стійкими до збоїв.

Незважаючи на беззаперечні переваги нативного коду, необхідність утримувати дві паралельні команди розробників для iOS та Android суттєво збільшує фінансові витрати підприємств та уповільнює час виходу продукту на ринок (time-to-market). У відповідь індустрія здійснила перехід до кросплатформних фреймворків, які у 2026 році досягли безпрецедентного рівня продуктивності, здатного конкурувати з нативними рішеннями. Домінуючими гравцями на ринку є Flutter, Kotlin Multiplatform (KMP) та React Native. Кожен з них має унікальну архітектурну філософію.

ПАРАДИГМИ КРОСПЛАТФОРМНОЇ РОЗРОБКИ: ГЛИБОКИЙ ТЕХНІЧНИЙ АНАЛІЗ

Незважаючи на беззаперечні переваги нативного коду, необхідність утримувати дві паралельні команди розробників для iOS та Android суттєво збільшує фінансові витрати підприємств та уповільнює час виходу продукту на ринок (time-to-market). У відповідь індустрія здійснила перехід до кросплатформних фреймворків, які у 2026 році досягли безпрецедентного рівня продуктивності, здатного конкурувати з нативними рішеннями. Домінуючими гравцями на ринку є Flutter, Kotlin Multiplatform (KMP) та React Native. Кожен з них має унікальну архітектурну філософію.

FLUTTER ТА ЙОГО ВЛАСНИЙ РУШІЙ РЕНДЕРИНГУ

Фреймворк Flutter від Google, що використовує мову програмування Dart, пропонує парадигму створення єдиної кодової бази як для бізнес-логіки, так і для візуального інтерфейсу. Головною інженерною особливістю Flutter є відмова від використання нативних віджетів операційної системи. Натомість Flutter відмальовує кожен

піксель на екрані самостійно, використовуючи власний високопродуктивний графічний рушій. Цей підхід гарантує абсолютну консистентність дизайну на всіх пристроях: додаток виглядатиме ідентично на будь-якій версії ОС. Швидкість розробки суттєво підвищується завдяки функції Hot Reload, яка компілює зміни в коді за мілісекунди. Однак Flutter вимагає використання так званих платформозалежних каналів (Platform Channels) для доступу до специфічних апаратних функцій, що може ускладнювати розробку складних системних рішень.

СТРАТЕГІЧНА ПЕРЕВАГА KOTLIN MULTIPLATFORM (KMP)

Абсолютно альтернативний підхід пропонує Kotlin Multiplatform (KMP). Його архітектурна філософія базується на принципі «спільна бізнес-логіка, нативний UI». Розробники пишуть ядро додатку (мережеві запити, взаємодію з локальними базами даних, складні алгоритми обробки інформації) мовою Kotlin, і цей код компілюється у нативні бінарні файли для кожної цільової платформи. Водночас користувацький інтерфейс створюється окремо нативними засобами кожної платформи (SwiftUI для iOS та Compose для Android).

Унікальною інженерною перевагою KMP є механізм використання ключових слів expect та actual. Це дозволяє розробникам створювати інтерфейс (очікувану поведінку — expect) у спільному модулі, а конкретну реалізацію (actual) прописувати безпосередньо для кожної платформи з прямим доступом до нативних API. Цей фреймворк став стандартом для великого корпоративного сектору (Enterprise) та FinTech систем, оскільки він дозволяє уникнути проблем із продуктивністю так званих «мостів» (bridges), які притаманні React Native.

ВИСНОВКИ

Глибокий аналіз парадигм програмування для мобільних платформ, що домінують у 2026 році, дозволяє констатувати перехід індустрії від екстенсивного написання коду до стадії зрілої, високоефективної програмної інженерії. Головним архітектурним трендом стало поєднання нативної продуктивності, зумовленої масовим переходом на декларативні інтерфейси (SwiftUI, Compose), з кросплатформною універсальністю. Дихотомія вибору сучасних фреймворків зводиться до балансування між швидкістю створення унікальних інтерфейсів (шлях Flutter) та розробкою масштабованих систем із нативним рівнем взаємодії та спільною бізнес-логікою (шлях Kotlin Multiplatform з його унікальним механізмом expect/actual).

Нові горизонти для розробників відкриває інтеграція з екосистемою IoT та перехід до периферійних обчислень (Edge Computing), що стало можливим завдяки мережам 5G. Паралельно з цим, управління великими кодовими базами корпоративного рівня вирішується шляхом впровадження мікрофронтендів та побудови Super Apps.

Нарешті, сучасні методи програмування стали нерозривно пов'язаними з аспектами безпеки. Впровадження практик DevSecOps, автоматизація CI/CD конвеєрів та суворе дотримання рекомендацій щодо бінарного захисту та криптографії згідно з OWASP Mobile перетворюють кібербезпеку з додаткової опції на фундаментальний критерій якості програмного продукту на етапі його проектування. Дослідження та презентація цих складних інженерних явищ вимагає від майбутніх спеціалістів з комп'ютерної інженерії найвищого рівня технічної та академічної грамотності.

Список літератури

5. Смірнова Т. В., Константинова Л. В., Конопліцька-Слободенюк О. К. та ін. Дослідження сучасного стану SIEM-систем. Кібербезпека: освіта, наука, техніка. 2024. № 1 (25). С. 6–18.

6. Босько В. В., Березюк І. А., Константинова Л. В., Конопліцька-Слободенюк О. К., Клюй А. Я. Аналіз та практична оцінка інструментів штучного інтелекту для веб-дизайну. Таврійський науковий вісник. Серія: Технічні науки. 2025. Вип. 1 (5). С. 65–73.

7. Снітко Ю. М. Огляд програмних середовищ для розробки мобільних додатків. Тези 76-ї наукової конференції (Полтава, 14–23 травня 2024 р.). Полтава : Нац. ун-т ім. Юрія Кондратюка, 2024. Т. 1. С. 482–483.

4. Дячук С. Ф., Борівець Б. Я. Крос-платформна розробка мобільних додатків за допомогою технології Xamarin. Матеріали VIII науково-технічної конференції. Тернопіль : ТНТУ, 2020. С. 131.

5. Kuznetsov, O., Smirnov, O., Akhmetov, B., Alimseitova, Z., & Imoize, A. L. (2025). Deep learning frontiers in copy-move forgery detection: Advances, challenges, and future directions. In Advancements in cybersecurity: Next-generation systems and applications

УДК 004.946, 355.40, 004.8

І.А. Лисенко, О.В. Лисенко
min_max@i.ua, ac74@i.ua
Центральноукраїнський національний технічний університет, Кропивницький

ЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СКЛАДОВОЇ В УМОВАХ СУЧАСНИХ КОНФЛІКТІВ НА ПРИКЛАДІ ДРУГОЇ СВІТОВОЇ ВІЙНИ ТА РОЗПАДУ СРСР

Інформаційна складова є одним із ключових доменів сучасних конфліктів поряд із воєнним, економічним, політичним та гуманітарним. У XXI столітті війна виходить за межі традиційного поля бою та поширюється у когнітивний простір, а також у сферу даних, комунікацій і кіберінфраструктури. За цих умов вирішального значення набуває не лише фізичний контроль територій, а й здатність впливати на сприйняття, інтерпретацію подій і поведінку цільових аудиторій.

Сучасні конфлікти характеризуються зростанням ролі інформаційних операцій, які охоплюють широкий спектр інструментів – від класичної пропаганди до кібероперацій, аналізу великих даних і застосування технологій штучного інтелекту (ШІ). Інформаційний вплив стає інтегрованою складовою стратегічного планування, поєднуючись із військовими, дипломатичними та економічними заходами.

Для розуміння трансформації інформаційної боротьби у стратегічний інструмент сучасних війн доцільно звернутися до історичних прикладів, зокрема подій Другої світової війни та процесів, пов'язаних із розпадом СРСР. Їх аналіз дозволяє простежити еволюцію інформаційних методів впливу – від масової пропаганди до складних багаторівневих систем інформаційного та когнітивного впливу, характерних для конфліктів XXI століття.

Інформаційна війна в роки Другої світової війни. Під час Другої світової війни [1] інформаційна складова вперше в історії набула безпрецедентного масштабу та системності. Засоби масової комунікації – радіо, кіно, друковані матеріали – використовувалися як інструменти впливу на масову свідомість, а розвиток криптографії та розвідувальних технологій суттєво підвищив ефективність інформаційного забезпечення військових операцій.

Криптографія та розвідка. Одним із ключових чинників інформаційної переваги союзників стало дешифрування німецької шифрувальної машини Енігма. У межах операцій ULTRA здійснювалося систематичне перехоплення й аналіз зашифрованих повідомлень, що дозволяло отримувати важливу розвідувальну інформацію, своєчасно реагувати на дії противника та підвищувати ефективність стратегічного планування.

Пропаганда та морально-психологічний вплив. Нацистська Німеччина сформувала централізовану систему пропаганди під керівництвом Йозефа Геббельса, яка активно використовувала медіа для формування лояльності населення та мобілізації ресурсів. У відповідь союзні держави (США, Велика Британія та СРСР) реалізовували власні інформаційні кампанії, спрямовані на підтримання морального духу, легітимацію воєнних дій і дискредитацію противника.

Інформаційне управління масами. Контроль над інформаційними потоками, цензура та координація медійного контенту дозволяли державам забезпечувати внутрішню стабільність, підтримувати суспільну мобілізацію та зменшувати вплив негативної інформації в умовах війни.

Дезінформація та стратегічне введення в оману. Важливу роль відігравали операції дезінформації, зокрема операція «Fortitude» («Фортеці»), яка була складовою ширшої кампанії з введення противника в оману перед висадкою союзників у Нормандії. Використання фальшивих радіоповідомлень, інсценованих військових формувань і контрольованих витоків інформації дозволило відвернути увагу німецького командування від реального напрямку наступу.

Досвід Другої світової війни засвідчив, що інформаційна складова є невід'ємною частиною військового протистояння, здатною забезпечувати стратегічні переваги. Саме в цей період було закладено основи сучасних інформаційних операцій, які у XXI столітті трансформувалися у комплексні багатовимірні системи впливу.

Інформаційна складова в контексті розпаду СРСР. У 1980-1990-х роках інформаційна сфера стала важливим чинником трансформацій, що супроводжували кризу та подальший розпад СРСР [2]. Лібералізація інформаційного простору, поява альтернативних джерел інформації та зміна характеру публічної комунікації суттєво вплинули на суспільно-політичні процеси, взаємодію між центром і республіками, а також на сприйняття державної влади.

Гласність і інформаційна відкритість. Реформи, ініційовані Михайлом Горбачовим, передбачали послаблення цензури та розширення доступу до інформації. Це сприяло появі нових медіа, публічному обговоренню раніше табуйованих тем (зокрема політичних репресій, економічних проблем, національних питань) і формуванню більш критичного ставлення до існуючої системи.

Інформаційна асиметрія та плюралізація наративів. У республіках СРСР поступово зростала кількість альтернативних інтерпретацій історії та сучасності. Поряд із офіційними джерелами інформації поширювалися

матеріали незалежних видань і зарубіжних медіа, що сприяло формуванню різних, інколи конкуруючих наративів щодо розвитку держави, її політики та перспектив.

Психологічні та соціальні ефекти інформаційних змін. Розширення інформаційного простору супроводжувалося зростанням рівня невизначеності, суспільної тривожності та критичного переосмислення попередніх ідеологічних установок. Це впливало на рівень довіри до державних інститутів і сприяло активізації суспільно-політичних процесів.

Інформаційний чинник у системі причин розпаду. Інформаційні процеси відіграли важливу роль у трансформації радянського суспільства, однак вони діяли у взаємодії з економічними, політичними та національними чинниками. Зниження контролю над інформаційним простором посилило відкритість суспільства, що, у поєднанні з внутрішніми кризовими явищами, сприяло дезінтеграційним процесам.

Досвід розпаду СРСР демонструє, що інформаційна складова може виступати потужним каталізатором суспільно-політичних змін, особливо в умовах трансформації політичної системи. Водночас її вплив є комплексним і реалізується у взаємодії з іншими структурними чинниками розвитку держави.

Сучасна війна – це змагання за розум, довіру та ідентичність. Інформаційна складова стала не допоміжною, а визначальною у гібридних війнах XXI століття [3-9]. Вона трансформувалася з інструмента супроводу у самостійний домен ведення війни, де вирішальним є контроль над сприйняттям, інтерпретацією подій та поведінкою аудиторій. Для успішної протидії інструментам інформаційному впливу у сучасних конфліктах, критично важливо їх виявити, зрозуміти та чітко визначити. Проаналізуємо на реальних прикладах форми інструментів інформаційного впливу у сучасних конфліктах.

Кіберпростір і соціальні мережі як поле бою нового покоління. Кіберпростір і соціальні мережі стали головним полем битви. Поширення дезінформації, фейків, маніпулятивних відео й бот-мереж створює ефект масового інформаційного тиску. Водночас сучасний етап характеризується активним використанням штучного інтелекту – зокрема deepfake, генеративних моделей тексту та голосу, що суттєво ускладнює верифікацію інформації та формує феномен «кризи довіри до фактів».

Приклад: під час війни Росії проти України (з 2014 р. і особливо після повномасштабного вторгнення 2022 р.) ведеться масштабна інформаційна кампанія: Росія намагається нав'язати власний наратив через пропаганду, тоді як Україна ефективно використовує цифрову дипломатію, OSINT-спільноти та соціальні мережі для мобілізації міжнародної підтримки. Додатково фіксуються приклади транснаціональних інформаційних атак, дезінформаційні кампанії, спрямовані на країни НАТО. Зокрема, провокація Росії проти Польщі у вересні 2025 р. Масована хвиля дезінформації напередодні 10 вересня – дати вторгнення 21 БПЛА на територію Польщі з поширенням також на соціальні мережі Франції, Німеччини та Румунії.

Наративна та когнітивна війна. Наративна війна ведеться за інтерпретацію подій, причин конфлікту, визначення «агресора» і «жертви». У сучасних умовах вона доповнюється когнітивною війною, яка спрямована не лише на зміну інформаційного поля, а й на трансформацію мислення, емоцій та поведінки індивідів і соціальних груп. У війні за «правду» перемога досягається через контроль над інформаційними потоками.

Приклад: під час війни у Сирії (2011-2024 р.р.) всі сторони – уряд, опозиція, Ісламська держава, а також міжнародні актори (США, РФ, Туреччина) – активно використовували інформаційні ресурси, включаючи візуальні докази (відео, фото), для впливу на громадську думку та легітимації своїх дій.

Кіберудари, інформаційний саботаж та контроль інфраструктури. Кіберудари та інформаційний саботаж є невід'ємною частиною сучасного конфлікту. Вони спрямовані на паралізацію державних структур, критичної інфраструктури та медіасистем. Важливою складовою стає також контроль інформаційної інфраструктури – блокування платформ, обмеження доступу до мережі, формування інформаційного вакууму.

Приклад: у війні в Нагірному Карабаху (2020 р.) між Вірменією та Азербайджаном поряд із застосуванням БПЛА активно використовувалися кібероперації та інформаційні кампанії. Азербайджан реалізував ефективну цифрову стратегію, що поєднувала військові дії з медійним впливом і міжнародною легітимацією. Росія (2026 р.) впроваджує блокування окремих платформ та мережі Інтернет в цілому.

Алгоритмічний вплив і data-driven пропаганда. Сучасна інформаційна війна дедалі більше базується на використанні алгоритмів соціальних платформ та аналізі великих даних. Це дозволяє здійснювати точковий вплив на цільові аудиторії через мікросегментацію, персоналізовані повідомлення та штучне формування інформаційних трендів.

Такі підходи реалізують концепцію data-driven propaganda, де рішення про інформаційні кампанії приймаються на основі аналітики поведінки користувачів.

Приклад: під час російсько-української війни (особливо після 2022 р.) російські інформаційні мережі активно використовували алгоритми соціальних платформ (Facebook, YouTube, TikTok) для просування дезінформаційного контенту через координовані мережі ботів і тролів. За рахунок масового поширення однотипних повідомлень і штучного підвищення їх популярності алгоритми рекомендацій підсилювали видимість такого контенту, формуючи викривлену інформаційну картину для окремих аудиторій у різних країнах. Паралельно застосовувалась сегментація користувачів (за мовою, регіоном, політичними поглядами) для адаптації пропагандистських меседжів, що дозволяло підвищити ефективність впливу на міжнародну громадську думку.

OSINT та counter-OSINT як новий вимір інформаційного протистояння. Відкриті джерела (OSINT) стали важливим інструментом документування подій війни (геолокація, аналіз відео, супутникові знімки). Водночас

розвивається counter-OSINT – заходи протидії, що включають маскування, створення фальсифікованих доказів і маніпуляцію метаданими.

Приклади: 1. Під час громадянської війни у Сирія (з 2011 р. по 2024 р.) OSINT-спільнота фактично сформувалася як окремий напрям аналітики. Дослідники та незалежні групи, зокрема Bellingcat та Conflict Intelligence Team (CIT), використовували відео з соціальних мереж, супутникові знімки та геолокацію для підтвердження застосування хімічної зброї, ідентифікації типів озброєння та встановлення місць ударів. Водночас сторони конфлікту активно застосовували counter-OSINT: інсценовані відео, маніпуляції з датами та географією публікацій, поширення суперечливих версій подій, а також обмеження доступу незалежних спостерігачів до зон бойових дій. Це призвело до формування складного інформаційного середовища, де перевірка даних вимагала багаторівневого аналізу та перехресної верифікації. Саме сирійський конфлікт став відправною точкою інституціоналізації OSINT як інструменту міжнародної журналістики та розслідувань, що згодом набув системного застосування у конфліктах нового покоління. 2. Під час російсько-української війни (з 2022 р.) OSINT-спільноти, ті ж Bellingcat та CIT, активно використовували геолокацію відео, супутникові знімки та аналіз відкритих даних для підтвердження переміщень військ, фактів руйнувань і воєнних злочинів. У відповідь російська сторона застосовувала counter-OSINT практики: приховування техніки, використання маскування, поширення інсценованих або змонтованих відео, а також маніпуляцію контекстом і часом публікацій. Це призвело до формування своєрідної «гонки верифікації», де кожна сторона намагається як підтвердити власні дані, так і дискредитувати інформацію противника. 3. У війні в секторі Гази (2023-2024 рр.) OSINT-аналітики використовували супутникові знімки та відео з соцмереж для перевірки наслідків ударів і встановлення їх географічної прив'язки. Водночас обидві сторони конфлікту вдавалися до counter-OSINT – обмеження доступу журналістів, затримки публікації інформації, а також поширення неповних або вирваних з контексту матеріалів, що ускладнювало незалежну перевірку подій.

Інформаційна мобілізація, деморалізація та меметична війна. Інформаційні кампанії використовуються для мобілізації власного населення та деморалізації противника. Важливу роль відіграє меметична війна, яка через прості, емоційно насичені образи швидко поширюється в цифровому середовищі та формує суспільні настрої.

Приклад: у війні в секторі Гази (2023-2024 рр.) інформаційна боротьба між сторонами конфлікту (Ізраїль і ХАМАС) велася у режимі реального часу через глобальні медіа та соціальні мережі, супроводжуючись високим рівнем емоційної поляризації аудиторій, змінюючи акценти щодо правомірності дій, масштабів втрат і причин ескалації.

Глобалізація інформаційних конфліктів. Сучасні конфлікти мають транснаціональний характер інформаційного впливу. Інформаційні операції виходять за межі безпосередніх учасників і спрямовуються на міжнародну аудиторію з метою формування вигідних наративів, зокрема щодо легітимності дій, демократичних цінностей або безпекових загроз.

Приклад: 1. Під час російсько-української війни (після 2022 р.) інформаційне протистояння вийшло далеко за межі Україна - Росія. Обидві сторони активно працюють із глобальною аудиторією: Україна використовує цифрову дипломатію, міжнародні медіа та соціальні мережі для формування підтримки з боку країн Заходу, тоді як Росія поширює альтернативні наративи через міжнародні інформаційні мережі, зокрема в країнах Африки, Азії та Латинської Америки. При цьому інформаційні кампанії адаптуються до регіональних особливостей аудиторії (історичний контекст, політичні настрої, соціально-економічні фактори), що свідчить про глобалізований і таргетований характер сучасної інформаційної війни. 2. У війні в секторі Гази (2023-2024 рр.) інформаційна боротьба між Ізраїль та ХАМАС швидко набула глобального масштабу. Контент із зони конфлікту миттєво поширювався через соціальні мережі, формуючи протилежні наративи в різних регіонах світу. У США та країнах Європи це спричинило політичні дискусії та масові протести, тоді як у країнах Близького Сходу й Глобального Півдня домінували інші інтерпретації подій. Таким чином, інформаційний конфлікт трансформувався у глобальне змагання за вплив на міжнародну громадську думку.

Сучасні інформаційні війни характеризуються глобалізацією наративів, де цільовою аудиторією стає не лише населення країн-учасниць, а й світова спільнота.

Гібридний характер конфлікту. Поєднання військових дій, пропаганди, економічного тиску, кібератак і правових маніпуляцій стало типовим для сучасних війн. Інформаційна складова інтегрується з іншими інструментами впливу, формуючи єдину багатовимірну стратегію.

Приклад: повномасштабне вторгнення РФ в Україну супроводжується: атаками на телекомунікаційну та урядову інфраструктуру; масштабними дезінформаційними кампаніями, як всередині Росії, так і за її межами; використанням ШІ-контенту та соціальних мереж; поєднанням військового, дипломатичного, енергетичного та інформаційного тиску.

Слід розрізняти інформаційні конфлікти за масштабом. Очевидно, що локальні інформаційні конфлікти за рядом критеріїв відрізаються від глобальних. Аналіз інформаційної складової конфліктів [3-9] дозволив чітко визначити відмінність за критеріями (табл. 1).

Таблиця 1

Порівняння локального та глобалізованого інформаційного конфлікту

Критерій	Локальний інформаційний конфлікт	Глобалізований інформаційний конфлікт
Масштаб впливу	Обмежений територією однієї держави або регіону	Транснаціональний, охоплює глобальну аудиторію
Цільова аудиторія	Населення країни або безпосередніх учасників конфлікту	Міжнародна спільнота, уряди, глобальні медіа, діаспори
Основна мета	Внутрішня мобілізація, деморалізація противника	Формування міжнародної підтримки, легітимація дій, вплив на глобальну політику
Канали комунікації	Національні ЗМІ, локальні соцмережі	Глобальні медіа, міжнародні платформи (X, YouTube, TikTok), дипломатичні канали
Характер наративів	Орієнтація на локальний контекст (історія, культура, внутрішня політика)	Адаптовані під різні регіони світу, мультинаративність
Роль соціальних мереж	Обмежена локальними аудиторіями	Ключова – алгоритмічне поширення контенту, віральність, глобальні тренди
Використання ШІ та Big Data	Обмежене або фрагментарне	Системне: microtargeting, data-driven пропаганда, генеративний контент
Алгоритмічний вплив	Мінімальний або неконтрольований	Цілеспрямоване використання алгоритмів для підсилення наративів
OSINT / counter-OSINT	Епізодичне використання	Масове застосування, «гонка верифікації» на глобальному рівні
Когнітивний вплив	Спрямований на локальне населення	Орієнтований на різні культурні та політичні групи у світі
Швидкість поширення інформації	Відносно повільна	Миттєва (реальний час, вірусне поширення)
Контроль інформаційного простору	Часто централізований (цензура, державні медіа)	Частково втрачений через глобальні платформи, але компенсується інформаційними операціями
Рівень дезінформації	Локалізований	Масштабний, багатомовний, адаптований під різні аудиторії
Меметична війна	Обмежена	Масова, глобальні меми як інструмент впливу

Глобалізація інформаційних конфліктів трансформує їх із локальних інструментів впливу на внутрішню аудиторію у складні багаторівневі системи, спрямовані на формування світового інформаційного порядку денного. Вирішальним стає не лише контроль інформації, а й здатність адаптувати її до різних культурних, політичних та когнітивних контекстів. На підставі аналізу [4-9] наведено приклади форм інформаційного впливу у сучасних конфліктах (табл. 2).

Таблиця 2

Форми інформаційного впливу у сучасних конфліктах

Конфлікт	Форми інформаційного впливу
Російсько-українська війна (з 2014 р. по тепер*)	Кіберудари; соцмережі; дезінформація; цифрова дипломатія; ШІ-генерований контент (deepfake); когнітивна війна; OSINT / counter-OSINT; алгоритмічне підсилення наративів; data-driven пропаганда; меметична війна; контроль інформаційної інфраструктури
Повномасштабна громадянська війна в Сирії (2011-2024 р.р.)	Медіа-пропаганда; інформаційні кампанії різних акторів; наративна війна; міжнародні інформаційні операції; когнітивний вплив; обмеження доступу до інформації; використання візуальних доказів (відео/фото) як інструменту впливу
Азербайджано-вірменська війна за Нагірний Карабах (2020-2023 р.р.)	Кампанії в соцмережах; візуальний PR (відео БПЛА); геополітична пропаганда; OSINT-аналіз і контрнаративи; інформаційна легітимація військових успіхів; алгоритмічне поширення контенту
Війна в секторі Гази між Ізраїлем і ХАМАС (2023-2024 р.р.)	Світова медіа-боротьба; дезінформація; маніпуляції втратами; емоційно орієнтовані кампанії; меметична війна; мобілізація через соцмережі; когнітивна поляризація аудиторії; інформаційні кризи в реальному часі
Конфлікт Ізраїлю з Іраном (2025 р.)	Кібератаки; кампанії дезінформації; пропаганда через медіа; інформаційна мобілізація населення; операції впливу на міжнародні наративи; ШІ-контент; кібершпигунство; атаки на критичну інфраструктуру; психологічні операції (PSYOPS)
Конфлікт між Пакистаном та Індією (2025 р.)	Кіберманіпуляції; медіа-кампанії; дипломатичний тиск через ЗМІ; ембарго інформаційних потоків; контроль інформаційного простору; наративна конкуренція; алгоритмічне поширення; інформаційна ескалація через соцмережі
Гібридна атака Росії на країни Європи члени НАТО (2025-2026 р.р.)	Соцмережі; дезінформація; кібердиверсії; інформаційні операції впливу; втручання у вибори; когнітивна війна; використання бот-мереж; energy & info leverage; стратегічні наративи
Операція США проти Венесуели (3 січня 2026 р.)	Інформаційно-психологічні операції, кампанії делегітимізації влади, кіберудари по державній інфраструктурі, використання соцмереж для формування протестних настроїв, міжнародні медіа-наративи щодо демократизації
Війна Ізраїлю та США проти Ірану (з 28 лютого 2026 р. по тепер *)	Масштабні кібератаки на критичну інфраструктуру, інформаційні кампанії глобального рівня, використання ШІ для створення deepfake-контенту, мобілізація населення через цифрові платформи, боротьба за контроль над міжнародними медіа-наративами

* на момент написання даного матеріалу 05.04.2026 р.

Історичні приклади Другої світової війни та розпаду СРСР демонструють, що інформація виступає одним із ключових стратегічних ресурсів у конфліктах. На сучасному етапі її роль суттєво зросла та трансформувалася: завдяки цифровим технологіям інформація стала не лише інструментом управління, а й самостійним доменом протиборства. У сучасних конфліктах боротьба ведеться не лише за території, а й за свідомість людей, за контроль над наративами та легітимацію дій на міжнародній арені.

Високотехнологічні засоби комунікації, інтернет і соціальні мережі забезпечують безпрецедентну швидкість поширення інформації, що створює нові виклики для безпеки, управління та стабільності держав. Ефективна інформаційна стратегія дозволяє державам формувати міжнародну підтримку, впливати на морально-психологічний стан противника, підтримувати довіру громадян і підвищувати стійкість суспільства до дестабілізаційних впливів.

Сучасні інформаційні війни характеризуються переходом від масової пропаганди до високоточних, персоналізованих і алгоритмічно підсилених впливів, що базуються на аналізі великих даних, застосуванні технологій штучного інтелекту та когнітивному впливі на поведінку індивідів і соціальних груп.

Отже, в умовах сучасних конфліктів інформаційна складова виступає критично важливою сферою, що суттєво впливає на перебіг і результати протистояння у взаємодії з військовими, економічними та політичними чинниками. Інформаційна перевага стає стратегічною необхідністю, оскільки дозволяє не лише досягати результатів у військовій сфері, а й формувати сприйняття реальності в масовій свідомості. У зв'язку з цим формування ефективної національної інформаційної політики, розвиток кібербезпеки, підвищення рівня інформаційної грамотності населення та вдосконалення стратегічних комунікацій мають розглядатися як пріоритетні напрями забезпечення національної безпеки.

Список літератури

1. Лисенко І. А., Лук'яненко Р. О. Дослідження методів ведення інформаційно-психологічних операцій використаних союзниками в другій світовій війні // Проблеми та шляхи захисту інформаційно-психологічної і духовної безпеки особи, суспільства, держави : матеріали V Міжнар. наук.- практ. конф., м. Київ, 28 червня 2024 р. – К., 2025. – С. 23. URL: https://dspace.kntu.kr.ua/handle/123456789/16653

2. Лисенко І. А., Паращенко Д. С. Аналіз внутрішніх політичних та економічних факторів, що сприяли розпаду СРСР // Проблеми та шляхи захисту інформаційно-психологічної і духовної безпеки особи, суспільства, держави : V Міжнар. наук.- практ. конф., м. Київ, 28 червня 2024 р. – К., 2025. – С. 22. URL: https://dspace.kntu.kr.ua/handle/123456789/16655

3. NATO Strategic Communications Centre of Excellence. Responding to Cognitive Security Challenges. Riga : NATO StratCom COE, 2019. 100 p. URL: https://stratcomcoe.org/cuploads/pfiles/web_Responing-to-Cognitive.pdf

4. Helmus T. C., Holynska K. Ukrainian Resistance to Russian Disinformation: Lessons for Future Conflict. Santa Monica, CA : RAND Corporation, 2024. 96 p. DOI: https://doi.org/10.7249/RRA2771-1

5. Bakirov A., Suleimenov I. Theoretical Bases of Methods of Counteraction to Modern Forms of Information Warfare // Computers. 2025. Vol. 14, No. 10. Article 410. DOI: https://doi.org/10.3390/computers14100410

6. Deppe C., Schaal G. Cognitive warfare: a conceptual analysis of the NATO ACT cognitive warfare exploratory concept // Frontiers in Big Data. 2024. Vol. 7. DOI: https://doi.org/10.3389/fdata.2024.1452129

7. Кучмій О., Фролова О. Використання соціальних медіа як інструменту сучасної гібридної війни // Acta de Historia & Politica : Saeculum XXI. – 2023. – Special Issue. – С. 93-104. DOI: https://doi.org/10.26693/ahpsxxi2023.si.093

8. Кириченко Ю. В., Сергієнко Т. І., Сластін В. О. Інформаційні війни як інструмент гібридної агресії: український досвід // Вісник Національного технічного університету України «КПІ». Політологія. Соціологія. Право. 2025. № 1(65). С. 89-95. DOI: https://doi.org/10.20535/2308-5053.2025.1(65).332563

9. Фещенко І. Інформаційна війна як органічна складова сучасного збройно-політичного конфлікту // Філософія та політологія в контексті сучасної культури. 2021. Т. 13, № 1. С. 96-103. DOI: https://doi.org/10.15421/352111

УДК 004. 42

В.О. Бабченко, *4 курс*
Науковий керівник: канд. тех.наук, ст. викл. Т. А Стабецька
babchenko.viktoriia1122@vu.cdu.edu.ua
Черкаський національний університет імені Богдана Хмельницького, Черкаси

АРХІТЕКТУРНІ ПІДХОДИ ДО РОЗРОБКИ ВЕБ-СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Сучасний етап розвитку інформаційних технологій характеризується стрімким поширенням електронної комерції, що обумовлює необхідність створення ефективних, масштабованих та надійних веб-систем. Архітектура таких систем відіграє ключову роль, оскільки визначає їхню продуктивність, гнучкість, здатність до масштабування та підтримки.

Актуальність дослідження полягає у необхідності вибору оптимальних архітектурних підходів до побудови веб-систем електронної комерції, які здатні забезпечити стабільну роботу при високому навантаженні, а також ефективну обробку транзакційних операцій. У сучасних умовах розробка подібних систем вимагає врахування як технічних, так і бізнес-вимог.

Метою роботи є аналіз та узагальнення сучасних архітектурних підходів до розробки веб-систем електронної комерції, а також визначення оптимальної архітектурної моделі для реалізації серверної частини онлайн-магазину.

У процесі дослідження було розглянуто основні архітектурні стилі, що застосовуються у веб-розробці, зокрема монолітну архітектуру, мікросервісний підхід та багатошарову архітектуру. Монолітні системи характеризуються простотою реалізації та централізованою структурою, однак мають обмеження щодо масштабування. Мікросервісна архітектура, у свою чергу, забезпечує високу гнучкість і незалежність компонентів, проте потребує складнішого управління та інфраструктури.

У результаті аналізу встановлено, що для більшості веб-систем електронної комерції доцільним є використання багатошарової архітектури, яка передбачає поділ системи на окремі рівні: рівень представлення, рівень бізнес-логіки та рівень доступу до даних. Такий підхід дозволяє забезпечити чітке розмежування відповідальностей між компонентами, підвищити підтримуваність системи та спростити її тестування.

Особливу увагу приділено організації взаємодії між шарами системи. Взаємодія типу «контролер – сервіс – репозиторій – база даних» дозволяє ефективно реалізувати бізнес-логіку та забезпечити контроль над виконанням транзакцій. Використання принципів об'єктно-орієнтованого програмування, зокрема SOLID, сприяє зменшенню зв'язності між компонентами та підвищує якість програмного коду.

Окремим аспектом дослідження є питання вибору системи управління базами даних для електронної комерції. Було проаналізовано особливості роботи реляційних СУБД у контексті транзакційного навантаження. Встановлено, що використання сучасних СУБД із підтримкою механізмів транзакцій та багатоверсійності дозволяє забезпечити узгодженість даних і коректну обробку одночасних запитів.

Важливим елементом архітектури є забезпечення безпеки системи. Реалізація механізмів автентифікації та авторизації, використання захищених протоколів передачі даних та контроль доступу до ресурсів дозволяють мінімізувати ризики несанкціонованого доступу та витоку інформації.

Також у роботі розглянуто питання інтеграції веб-системи з зовнішніми сервісами, зокрема платіжними системами. Використання API сторонніх сервісів дозволяє розширити функціональність системи та забезпечити реалізацію повного циклу онлайн-продажу.

У результаті проведеного дослідження визначено, що оптимальним підходом до побудови веб-систем електронної комерції є поєднання багатошарової архітектури з використанням сучасних принципів проєктування та інструментів розробки. Такий підхід дозволяє забезпечити баланс між продуктивністю, масштабованістю та складністю реалізації системи.

Отримані результати можуть бути використані під час розробки інформаційних систем електронної комерції, орієнтованих на обробку транзакційних операцій, а також при проєктуванні програмних рішень із високими вимогами до надійності та безпеки.

UDC 004.932:004.8:004.93

Oleh Breslavskyi[1], Oleksandr Dorenskyi[2], Dmytro Uhryn[1], Yurii Ushenko[1]
breslavskyi.oleh@chnu.edu.ua, dorenskyiop@kntu.kr.ua, d.ugryn@chnu.edu.ua, y.ushenko@chnu.edu.ua
[1]*Yuriy Fedkovych Chernivtsi National University, Chernivtsi*
[2]*Central Ukrainian National Technical University, Kropyvnytskyi*

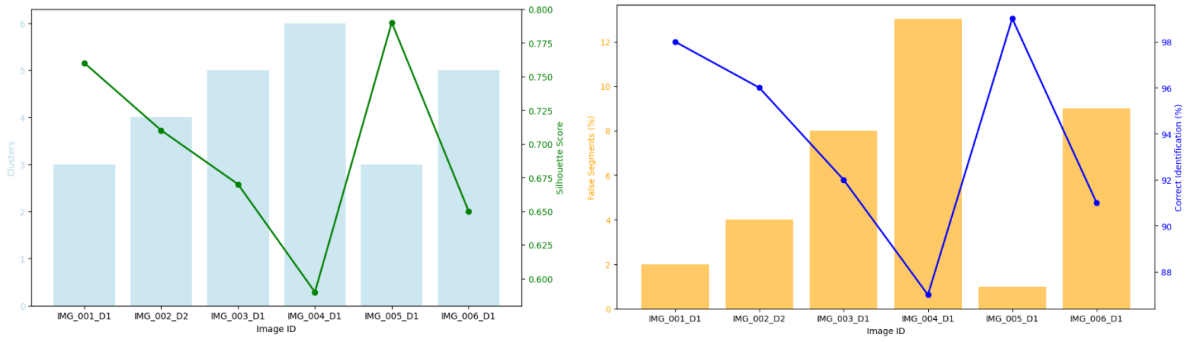
EXPERIMENTAL ANALYSIS OF THE IMPACT OF ADAPTIVE PREPROCESSING ON KEY FEATURE DETECTION AND HUMAN IDENTIFICATION ACCURACY IN COMPUTER VISION SYSTEMS

In the context of the rapid development of digital technologies and computer vision systems, particular attention is given to the tasks of automated analysis of digital images, particularly object detection, segmentation, and identification [1, 2]. These tasks, such as video surveillance, traffic safety, and public order maintenance, are primarily focused on human recognition. As is well known, this often takes place under conditions of limited visibility: fog, low lighting, reduced contrast, insufficient illumination, and so on. As explored in [1], for neural networks to make accurate decisions during classification, the extraction of key features of the object is critical. Under poor visibility conditions, these features are distorted, which leads to the need for improving recognition accuracy and reducing errors. This issue was addressed in the study [2] by implementing adaptive preprocessing, descriptor analysis, and deep learning models. The improvement of the accuracy and robustness of automated segmentation, clustering, and human identification in digital images under reduced visibility conditions is achieved by integrating adaptive preprocessing, feature analysis, and deep learning models to ensure the stable operation of computer vision systems through the enhancement of segmentation quality, detection, clustering, and human identification accuracy [2].

The aim of this study is to analyze the impact of adaptive preprocessing of digital images using classical methods and deep learning models on the quality of key feature extraction (specifically using the Silhouette Score metric) and its influence on the overall accuracy of human identification.

For the experimental study of the comprehensive combination of adaptive image preprocessing, feature descriptor analysis, and deep learning models (such as U-Net, Mask R-CNN, YOLOv8) [2], a sample of 350 digital images was created. This sample includes both proprietary photographs and open datasets, such as COCO and CrowdHuman. These images cover various scenes: individual persons, groups, and crowds, with the presence of external objects. Artificial degradation effects such as fog, noise, dimming, and reduced contrast were simulated. Additionally, normalization, γ-correction, CLAHE, Dehazing, and various filtering techniques (Median, Gaussian, Bilateral) were applied to compensate for distortions.

The experimental analysis of object clustering results after image preprocessing [2] (evaluating the impact of filtering on the Silhouette Score and human identification accuracy in digital images) showed that the application of adaptive preprocessing enhances the accuracy in computer vision systems. A comparison of metrics for each image in the context of clustering and identification is presented in Figure 1.



a) *b)*

Fig. 1. Comparative analysis of the number of clusters and Silhouette Score values for the test sample of digital images: a) number of clusters and Silhouette Score values for each image; b) percentage of false segments and correct identification.

The results of the experimental analysis showed that the highest classification result does not depend on the maximum number of clusters but on their optimal selection combined with quality preprocessing. The best results (correct human identification) were observed for image IMG_005_D1, where the optimal number of clusters is 3, and the percentage of correct identification reaches 99%. At the same time, images with a higher number of clusters, such as IMG_003_D1 (5 clusters), achieved 92% correct identifications. This pattern is confirmed by the mathematical quality assessment: the highest Silhouette Score is achieved for image IMG_005_D1, which also confirms the best clustering quality, particularly regarding identification accuracy. A low Silhouette Score indicates a deterioration in clustering

quality, reducing the correctness of identification. The smallest number of false segments was observed for images IMG_005_D1 and IMG_001_D1, which indicates the high effectiveness of image preprocessing for these scenes. Images with high degradation levels show low Silhouette Scores and a high number of false segments, confirming the challenges in clustering under significant image distortion.

It is worth noting that the quality restoration of key features in the image through adaptive filtering has allowed for the effective application of not only clustering methods but also deep learning models. Specifically, experiments demonstrated consistently high segmentation performance using the U-Net and Mask R-CNN architectures (IoU metric up to 0.95) and detection with the YOLOv8 model (mAP up to 0.95) [2]. This confirms the synergistic effect of integrating classical preprocessing and neural network approaches.

The advantages of the proposed approach lie in this synergistic integration, combining classical computer vision algorithms with deep learning models. Classical methods provide interpretability, high speed, and low computational cost, while neural networks offer a high ability to generalize and detect complex patterns. Additionally, the approach's adaptability to types of degradation is a key strength: rather than applying universal processing for each type of distortion (fog, dimming, impulse noise), a specialized combination of methods is used. This targeted approach allowed for a significant increase in the integral image quality index under the most challenging conditions. High accuracy and robustness are also achieved thanks to the comprehensive data preparation and the use of powerful architectures such as U-Net, Mask R-CNN, and YOLOv8.

However, the proposed approach has its limitations. One of the key challenges is the difficulty of compensating for smoke. The most challenging scenes for segmentation and detection remain those with smoke or combined effects. For scenes with smoke, the least improvement effect was observed due to the physical complexity of fully compensating for the semi-transparent layer in the image. The method is also sensitive to domain changes: for night-time detection, where the human silhouette is weakly expressed, it is often necessary to use background context and attention mechanisms. However, such solutions are sensitive to changes in cameras and scenarios, and thus require complex domain validation, rather than being tested on a single training dataset. Another unresolved issue is the transfer of features to other sensors, as training light-invariant features has certain limitations, making them difficult to transfer across different sensor types. Moreover, strict evaluation protocols are required to avoid the loss of the system's ability to distinguish objects.

The results obtained indicate that adaptive preprocessing and the correct selection of the number of clusters improve clustering efficiency and human identification accuracy, particularly in challenging conditions of digital image degradation. The integration of adaptive filtering methods during preprocessing helps mitigate the impact of complex visual environments, ensuring high reliability of feature description and subsequent object identification. Thus, the quality of digital images improves, segmentation and detection accuracy increases, and clustering efficiency is enhanced. As a result, human recognition and identification accuracy in digital images has increased to 99%, even under difficult visual conditions. The computer vision system has gained high robustness (resilience) to various types of image degradation and can be successfully and reliably applied in real-world video monitoring systems, ensuring traffic safety, public order, and more.

The prospects for further research should be directed towards addressing unresolved challenges in human identification under limited visibility conditions. In particular, there is a need to improve algorithms for handling the most complex types of image degradation, such as scenes with smoke (where full compensation of the semi-transparent layer remains difficult), and combined effects. Furthermore, an important step will be the thorough validation of the proposed solutions "across domains" to ensure their robustness to camera changes and various scenarios. Tasks related to uncertainty modelling, transferring recognition results beyond conventional (e.g., road) domains, and improving instance segmentation in scenes with dense crowds also remain relevant. Additionally, attention should be given to adapting the developed methods for use with various sensor types and establishing strict evaluation protocols to avoid the loss of discriminability during model training.

References

1. Dorenskyi O., Drieiev O. Drieieva H. The method of identifying key elements of a digital image in the decision-making process of classification by a neural network. Computer systems and information technologies. 1, 2026, P. 145-155. DOI: https://doi.org/10.31891/csit-2026-1-14.

2. Угрин Д. І., Доренський О. П., Ушенко Ю. О., Бреславський О. І. Методи та моделі інтелектуального комп'ютерного зору для ідентифікації й оцінки функціонального стану людини в умовах обмеженої видимості. Центральноукраїнський науковий вісник. Технічні науки, 13(44), С. 33-40. https://doi.org/10.32515/2664-262X.2026.13(44).33-40.

УДК 004.738.5:004.45

Д.О. Гребенюк, *ст. 3-го курсу*
Науковий керівник: В.А. Резніченко, *викладач*
denis.hrebeniuk2006@gmail.com, upsbilly@meta.ua
Центральноукраїнський національний технічний університет, Кропивницький

АНАЛІЗ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ ПРОТОКОЛУ WEBSOCKET У ФРЕЙМВОРКУ FASTAPI ДЛЯ СИСТЕМ РЕАЛЬНОГО ЧАСУ

У сучасній веб-розробці миттєвий обмін даними з користувачем перетворився з опціональної можливості на індустріальний стандарт. Користувачі очікують миттєвих оновлень у таких системах, як онлайн чати, платформи для трансляцій спортивних матчів, навігаційні застосунки, фінансові додатки, тощо [1]. Згідно зі звітом за 2025 рік, понад 68% нових веб-застосунків використовують WebSocket або аналогічні протоколи для покращення продуктивності передачі даних [2]. Вибір протоколу зв'язку безпосередньо впливає на швидкість, стабільність, масштабованість програмного забезпечення та загальне задоволення користувачів від системи.

Сучасна індустрія демонструє масовий перехід провідних технологічних компаній на використання WebSocket. Зокрема, популярні комунікаційні платформи (WhatsApp Web, Facebook Messenger, Slack, Discord) покладаються на цей протокол для миттєвої доставки повідомлень, індикації набору тексту та статусів присутності. Стрімінгові гіганти (YouTube, Twitch, Netflix) застосовують його для синхронізації чатів під час прямих трансляцій та миттєвого надсилання сповіщень. Крім того, інструменти спільної розробки (Figma, GitHub) та сервіси з динамічним ціноутворенням (Uber, Airbnb) використовують WebSocket для забезпечення безперервної синхронізації даних між клієнтами без необхідності ручного оновлення сторінки або застосунку [1].

Традиційно для отримання оновлень використовувався метод «опитування», при якому клієнт періодично надсилає HTTP-запити до сервера. Проте цей підхід має суттєві обмеження:

- Затримка. Оновлення відображаються лише після наступного інтервалу опитування, що призводить до отримання застарілої інформації.

- Неефективність смуги пропускання. Багаторазові HTTP-запити змушують систему постійно передавати надлишкові заголовки, що витрачає ресурси мережі.

- Навантаження на сервер. Високочастотне опитування від тисяч користувачів може критично перевантажити серверну інфраструктуру.

На противагу цьому, WebSocket - це сучасний протокол, що забезпечує постійне двостороннє з'єднання між клієнтом і сервером. Після ініціалізації через стандартний HTTP-запит («рукостискання») з'єднання перемикається на WebSocket, що дозволяє обом сторонам миттєво обмінюватися даними в режимі повного дуплексу.

Таблиця 1

Порівняльна характеристика методів передачі даних [3]

Характеристика	**HTTP**	**WebSockets**
Тип з'єднання	Повторювані HTTP-запити	Постійне TCP-з'єднання
Потік даних	Напівдуплексний	Двонаправлений
Затримка	Залежить від інтервалу	Майже миттєва
Використання смуги	Високе, через заголовки	Низьке, лише корисні дані
Масштабованість	Обмежена	Висока
Складність	Легко реалізувати	Потребує підтримки бекенду
Ідеальні сценарії	Рідкісні оновлення	Високочастотні дані

Експериментальні дані підтверджують, що при передачі великої кількості малих порцій даних (понад 100 повідомлень) використання WebSocket дозволяє підвищити продуктивність системи на кількасот відсотків порівняно з HTTP. Важливою особливістю WebSocket є його незалежність від апаратного забезпечення клієнта - протокол однаково ефективно працює як на сучасних, так і на застарілих комп'ютерах [4].

Важливим фактором, що суттєво знижує продуктивність HTTP у системах реального часу, є вплив надлишкових заголовків на розмір пакетів даних. Кожен HTTP-запит супроводжується обов'язковою передачею службової інформації. Дослідження показують, що при штучному збільшенні кількості полів заголовка (наприклад, до 100 додаткових параметрів) загальна швидкість передачі даних падає приблизно на 20%. У випадку WebSocket ця вразливість повністю відсутня - після успішного встановлення з'єднання корисне навантаження передається у вигляді компактних фреймів, де розмір службових метаданих зведений до мінімуму. Це робить протокол ідеальним для сценаріїв, де необхідно стабільно передавати сотні або тисячі дрібних повідомлень на секунду

Також дослідження впливу шифрування TLS, тобто перехід на HTTPS та WSS протоколи, продемонстрували, що воно має незначний вплив на швидкість обох протоколів при передачі даних. Це дозволяє впроваджувати захищені з'єднання в реальному часі без суттєвих втрат продуктивності.

Використання FastAPI у розробці Real-Time систем є обґрунтованим завдяки його асинхронним можливостям через бібліотеку asyncio. Фреймворк забезпечує нативну підтримку WebSocket та автоматичну валідацію даних через Pydantic. Асинхронність дозволяє серверу підтримувати тисячі активних TCP-з'єднань одночасно, споживаючи мінімальну кількість системних ресурсів, що є критичним для масштабованості великих IT-проектів [5].

Важливим аспектом розробки систем реального часу є захист даних. На відміну від стандартних REST API, де токени доступу (наприклад, JWT) передаються у заголовках кожного HTTP-запиту, при використанні WebSocket автентифікація зазвичай відбувається на етапі початкового HTTP-рукостискання. Фреймворк FastAPI, завдяки системі впровадження залежностей, Dependency Injection, дозволяє безперешкодно перевіряти легітимність користувача ще до того, як з'єднання буде остаточно прийнято сервером. Це унеможливлює несанкціонований доступ до потокових даних та захищає систему від витрат ресурсів через фішингові підключення.

Проведене дослідження підтверджує, що протокол WebSocket є безальтернативним вибором для розробки систем, які потребують високої швидкості передачі даних та масштабованості. Хоча традиційне опитування залишається валідним для архітектур із низькою частотою оновлень, перехід до подійно-орієнтованих систем на базі FastAPI та WebSockets є глобальним трендом розробки. Важливо підкреслити, що інтеграція механізмів автентифікації на етапі початкового «рукостискання» та використання захищених протоколів WSS дозволяє забезпечити необхідний рівень безпеки без значних втрат продуктивності. Таким чином, поєднання асинхронних можливостей FastAPI з повнодуплексними каналами зв'язку WebSocket створює надійний та захищений фундамент для побудови сучасних високонавантажених сервісів реального часу.

Список літератури

1. What are WebSockets used for?. Ably Realtime. URL: https://ably.com/topic/what-are-websockets-used-for (дата звернення: 08.04.2026).

2. Łasocha W., Badurowicz M. Comparison of WebSocket and HTTP protocol performance. Journal of Computer Sciences Institute. 2021. Vol. 19. P. 67–74. URL: https://doi.org/10.35784/jcsi.2452 (дата звернення: 08.04.2026).

3. Real-Time WebSockets vs Traditional Polling: Performance Comparison. Medium. URL: https://kodekx-solutions.medium.com/real-time-websockets-vs-traditional-polling-performance-comparison-b7230091d70e (дата звернення: 08.04.2026).

4. O'Riordan M. WebSocket vs HTTP: When to Use Each Protocol. WebSocket.org. URL: https://websocket.org/comparisons/http/ (дата звернення: 08.04.2026).

5. WebSockets - FastAPI. FastAPI. URL: https://fastapi.tiangolo.com/advanced/websockets/ (дата звернення: 08.04.2026).

УДК 004.77

Л. В. Константинова
liliyashel1976@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький, Україна

ДОСЛІДЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АРХІТЕКТУРНОЇ ЕФЕКТИВНОСТІ ТА ФУНКЦІОНАЛЬНИХ ОБМЕЖЕНЬ PWA

Розробка програмного забезпечення в періоди технологічних змін потребує також змін і нових підходів. Розвиток технологій у сфері вебпрограмування та нові потреби користувачів вимагають гнучких методів для швидкої роботи, зручності та можливостей працювати офлайн. Великою популярністю у виконанні багатьох потреб користується технологія PWA (Progressive Web Apps) і тому буде актуальним дослідити технологію та проаналізувати недоліки та переваги її застосування.

PWA (Progressive Web Apps) - це такий тип вебзастосунків, що поєднує і найкращі можливості вебпрограмування, і звичайних додатків [1]. Ця технологія підходить для створення webзастосунків, які вимагають швидкої взаємодії з користувачем, можливості працювати офлайн і доступу до функціонала, як у звичайних мобільних та комп'ютерних додатках. Використання PWA охоплює все більше сфер таких як: електронна комерція, стрімінгові сервіси та медіа, освітні платформи, інтернет-магазини, банківські послуги, комунікації і багато іншого. Найвідоміші з них: AliExpress, Twitter Lite, Strbucks, Spotify та ін.

Технологія PWA має ряд переваг [2]. Здатність працювати в автономному режимі, підтримка функції, такої як push-сповіщення дають PWA властивість подібності до нативних додатків для користувача, не потребуючи при цьому інсталяції з магазинів програм. Крім того ефективні й швидкі методи розробки, як перевага PWA, що робить щвидшим процес тестування.

PWA має деяку архітектурну перевагу перед нативними додатками. Вона забезпечується завдяки таким компонентам [2, 3, 4]:

1. Технологія Service Worker.
2. Архітектура App Shell Model.
3. Файл Web App Manifest.

Service Worker працює як прошарок між браузером та сервером, дозволяючи PWA функціонувати в автономному режимі, кешувати ресурси для швидкого доступу в режимі офлайн, обробляти push-сповіщення та багато іншого. Service Worker надає розробникам потужний інструмент для створення більш надійних і швидких веб-додатків з більш багатим досвідом користувача, особливо в режимі офлайн або з обмеженим інтернет-з'єднанням.

App Shell – архітектура, при якій оболонка сторінок PWA завантажується в кеш пристрою під час першого відвідування. В подальшому використанні програми каркас сторінок береться з локального кешу, а з сервера буде завантажено тільки контент. Ця архітектура робить інтерфейс PWA порівнянним за швидкістю з нативними додатками.

JSON-файл, Web App Manifest, містить інформацію про програму (метадані PWA). Застосовується замість створення застосунків під кожну ОС шляхом створення єдиної кодової бази, що адаптується під всі платформи через маніфест. Він розташовується в корні проєкта та підключається в HTML-файлі.

Для визначення продуктивності вебзастосунку до та після впровадження PWA проводились тестування [2], за результатами яких видно, що після впровадження - продуктивність застосунку покращилася. Крім того PWA може зменшити розмір переданих ресурсів та пришвидшити час завантаження вебзастосунку. Результати тестувань свідчать про те, що впровадження технології PWA на веб-сайті може покращити його продуктивність.

Як зазначається автором у [4] покращення продуктивності стало важливим проривом для PWA у 2024–2025 роках. PWA досягають часу завантаження протягом 2–3 секунд у мережах 3G, що відповідає або дає кращий результат ніж у нативних програм.

Було виявлено причини, чому підприємства переходять на PWA [3]:

1. Мобільний світ (понад половина веб-трафіку припадає на мобільні пристрої).
2. Зростання витрат на розробку (підтримка існуючих платформ є дорогою).
3. Терміновість використання (навіть мізерна затримка може призвести до втрати конверсій); PWA здатні швидко завантажуватися та працювати офлайн.
4. Присутність у пошуку (PWA індексуються в результатах пошуку, на відміну від нативних додатків).

Цей зсув не лише технологічний, а й стратегічний. PWA скорочують час виходу на ринок, уніфікують цифровий досвід та забезпечують вимірювану рентабельність інвестицій.

В роботі були означені ключові характеристики, що відмічають у прогресивних вебзастосунках:

1. Прогресивність. Працює безперебійно в різних браузерах та на різних пристроях, покращуючись завдяки новим можливостям [2, 3].

2. Адаптивність. Швидко адаптується до будь-якого розміру чи орієнтації екрана[3].

3. Незалежність від підключення. Кешує ресурси, щоб залишатися функціональним офлайн або в зонах з низьким рівнем мережевого зв'язку [1, 3].

4. Введення, подібне до додатків. Пропонує анімацію, жести та безперебійну навігацію [3, 4].

5. Безпека та захищеність. Працюючи через HTTPS, забезпечує безпеку даних.

6. Видимість. Контент індексується пошуковими системами, розширюючи видимість та охоплення.

7. Повторна взаємодія. Використовує push-сповіщення для повторного залучення користувачів своєчасними оновленнями.

8. Встановлення: Додається на головний екран пристрою без потреби встановлення з магазину.

9. Посилання. Поширюється просто URL-адресою.

Під час впровадження PWA необхідно враховувати особливості взаємодії з користувачем. Хоча PWA можуть досягти продуктивності, подібної до нативної, інтерфейс має бути розроблений, щоб він був знайомим користувачам мобільних пристроїв. Тобто повинно бути реалізовано навігація жестами, компонентів, оптимізованих для сенсорного керування, та переходи у звичайному стилі. Мета полягає в тому, щоб створити настільки безперебійний досвід, і зручність для користувачів.

Основні функціональні недоліки технології PWA заключаються в обмеженому доступі до апаратного забезпечення (Hardware), деяких обмеженнях на iOS, неможливість виконувати деякі складні завдання у фоновому режимі так ефективно, як звичайні застосунки, складні задачі 3D графіки, редагування відео, аудіо в PWA працюватиме повільніше.

PWA здатні обробляти приблизно 80–85% поширених випадків використання мобільних застосунків [4], включно з офлайн-функціональністю, push-сповіщеннями та інтеграцією пристроїв. Проте є ряд застосунків, що потребують доступу до апаратного забезпечення пристрою для, наприклад, редагування фото професійно, AR/VR чи ігри. Подібні завдання можуть лишатися у числі схильних до нативної розробки, хоча кількість випадків придатних для застосування PWA зростає з розширеннями можливостей браузерів.

Висновок: Статистичні дані показують, що 53% відвідувачів залишають веб-сторінку, якщо вона завантажується більше 3 секунд, що наголошує, що швидкість та адаптивність важливі фактори, для розробки програмних продуктів. Хоча нативні застосунки звичні своїм користувацьким інтерфейсом, широкою інтеграцією пристроїв, високою вартістю розробки та підтримки, PWA проявляють продуктивність схожу до нативних, але є швидшими та доступними в розробці. PWA вдале рішення для бізнесу, який прагне покращити свою цифрову присутність та зручність. Перехід до PWA-розробки вимагає стратегічного підходу, що мінімізує ризики та максимізує довгострокові вигоди. Найуспішніші впровадження можуть починатися з гібридної стратегії, де PWA спочатку доповнюють, а не замінюють існуючі нативні застосунки.

Список літератури

1. Що таке PWA-додатки - розробка Progressive Web Application, 21.10.24. URL: https://dan-it.com.ua/uk/blog/shho-take-pwa-dodatky-rozrobka-progressive-web-application/ (дата звернення: 5.04.26).

2. Ahyar Muawwal, The Implementation of PWA (Progressive Web App) Technology in Enhancing Website Performance & Mobile Accessibility. ResearchGate, Buletin Pos dan Telekomunikasi Vol. 22 No.1 (2024) URL: https://www.researchgate.net/publication/381388064_The_Implementation_of_PWA_Progressive_Web_App_Technology_in_Enhancing_Website_Performance_Mobile_Accessibility_The_Implementation_of_PWA_Progressive_Web_App_Technology_in_Enhancing_Website_Performance (дата звернення: 6.04.26).

3. A Complete Guide to Progressive Web Application Development for Modern Enterprises**,** Medium Engineering (2026). URL: https://medium.com/@pratheeshr.pr/a-complete-guide-to-progressive-web-application-development-for-modern-enterprises-ffe87c40a449 (дата звернення: 7.04.26)

4. Devin Rosario, Why Progressive Web Apps Will Beat Native in 2026. (2025) URL: https://gemini.google.com/app/05dbc4176b2f97bc?utm_source=app_launcher&utm_medium=owned&utm_campaign=base_all (дата звернення: 8.04.26)

УДК 004.75

Папіж Л. М., *аспірант*
факультет інформаційних систем та технологій,
ПВНЗ "Європейський університет", Київ
lmpapizh@gmail.com
Науковий керівник: Улічев О.С., *к.т.н., доцент*
Центральноукраїнський національний технічний університет, Кропивницький

СУЧАСНИЙ СТАН ТА НАУКОВО-МЕТОДОЛОГІЧНІ ЗАСАДИ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Сучасний етап розвитку програмної інженерії характеризується якісною трансформацією вимог до програмного забезпечення, що проявляється у підвищенні стандартів його надійності, функціональної коректності та безпечності експлуатації. Зазначена тенденція обумовлює необхідність поглибленого теоретико-методологічного осмислення процесів тестування як невід'ємної складової забезпечення якості програмних систем. Ускладнення архітектурних моделей програмних продуктів, скорочення їх життєвого циклу, а також імплементація гнучких і безперервних підходів до розробки об'єктивно зумовлюють перегляд традиційного уявлення про тестування як про завершальний етап життєвого циклу програмного забезпечення [1]. У сучасних умовах тестування функціонує як інтегрований процес, що пронизує всі стадії створення, постачання та експлуатації програмних систем, забезпечуючи їхню відповідність заданим вимогам і обмеженням.

У цьому контексті категорія тестованості програмного забезпечення набуває статусу однієї з базових наукових категорій, що характеризує властивості програмної системи у площині її придатності до верифікації та валідації. Концептуалізація тестованості дозволяє встановити закономірності взаємозв'язку між архітектурними характеристиками програмного продукту, структурною організацією програмного коду, рівнем формалізації вимог та ефективністю процедур виявлення дефектів [2; 3]. У зв'язку з цим тестованість доцільно визначати як інтегральну системну характеристику, що відображає здатність програмного забезпечення до відтворюваної, контрольованої та економічно обґрунтованої перевірки його функціональних і нефункціональних параметрів.

У межах сучасного наукового дискурсу тестованість інтерпретується як результат взаємодії сукупності структурних і функціональних характеристик програмної системи. Її формування детермінується рівнем модульності архітектури, ступенем зв'язності компонентів, прозорістю програмної логіки, однозначністю визначення вхідних і вихідних параметрів, а також наявністю формалізованих моделей поведінки системи [2; 3]. З огляду на це тестованість не може бути редукована до окремої технічної ознаки, а має розглядатися як багатовимірна характеристика, що піддається цілеспрямованому регулюванню через оптимізацію архітектурних рішень, удосконалення специфікацій та впровадження формалізованих і автоматизованих засобів тестування.

Структурно-логічну послідовність дослідження тестованості програмного забезпечення узагальнено на рис. 1, де відображено взаємозв'язок між чотирма ключовими змістовими блоками: визначенням тестованості як базової властивості програмних систем, формалізацією ролі тестування у забезпеченні якості та надійності, оцінюванням впливу цифрової трансформації на процедури перевірки, а також введенням методологічних рекомендацій.

Функціональна значущість тестованості проявляється у її системоутворювальній ролі в межах життєвого циклу програмного забезпечення. Вона визначає ефективність раннього виявлення дефектів, впливає на складність і ресурсоємність регресійного тестування, а також забезпечує можливість інтеграції автоматизованих процедур перевірки у процеси безперервної інтеграції та постачання. У цьому зв'язку тестованість доцільно розглядати як індикатор рівня технологічної зрілості програмного продукту, що відображає ступінь відповідності його архітектурних і функціональних характеристик сучасним вимогам забезпечення якості. Таким чином, тестованість постає не лише як об'єкт технічного аналізу, а як концептуальна категорія, що визначає методологічні засади організації процесів тестування в умовах сучасної програмної інженерії.

У сучасному науково-інженерному дискурсі тестування виходить за межі вузького розуміння як процедури виявлення дефектів і постає як багатофункціональний механізм підтвердження відповідності програмного продукту встановленим вимогам, специфікаціям та умовам експлуатації, а також як інструмент ідентифікації потенційних ризиків, здатних зумовити критичні відмови або порушення інформаційної безпеки [4]. При цьому інтеграція різних типів тестування, зокрема функціонального, інтеграційного, приймального, навантажувального та юзабіліті-тестування формує комплексну багатовимірну модель оцінювання якості програмної системи, що охоплює її функціональну коректність, стабільність функціонування та експлуатаційну придатність у реальному середовищі.

Рис. 1. Структурно-логічна модель аналізу тестованості ПЗ та етапів її дослідження

Емпіричні дані, отримані в межах функціонування великих IT-компаній, фінансових установ, телекомунікаційних систем, а також у транспортній і медичній галузях, підтверджують визначальну роль системно організованого тестування у забезпеченні відмовостійкості, захищеності даних та передбачуваності поведінки програмних компонентів в умовах реальних навантажень. Зокрема, впровадження автоматизованого регресійного тестування та комплексного навантажувального аналізу дозволяє істотно знизити частоту критичних інцидентів у продуктивному середовищі, мінімізувати час простоїв та підвищити якість надання послуг кінцевим користувачам. Водночас аналіз негативних кейсів, зокрема у транспортних та медичних інформаційних системах, свідчить про те, що недостатня глибина валідаційних процедур, відсутність моделювання граничних і критичних сценаріїв функціонування, а також відсутність формалізованих підходів до оцінювання людино-машинної взаємодії можуть призводити не лише до економічних втрат, а й до безпосередніх загроз безпеці людини.

У цьому контексті тестування доцільно концептуалізувати як інструмент управління ризиками, що функціонує у межах життєвого циклу програмного забезпечення. Його сутнісна функція полягає не лише у фіксації дефектів, а у формуванні керованого середовища ризиків, у якому потенційні відмови підлягають прогнозуванню, локалізації та мінімізації за допомогою цілеспрямовано спроєктованих процедур верифікації та валідації. У зв'язку з цим ефективні практики тестування ґрунтуються на поєднанні формалізованих стратегій із інструментальною підтримкою автоматизації, що забезпечує відтворюваність, масштабованість і оперативність виконання тестових процедур. Для високоризикових предметних областей це зумовлює перехід до ризик-орієнтованих моделей тестування, у межах яких пріоритезація перевірок здійснюється з урахуванням критичності функціональних компонентів, імовірності виникнення відмов та масштабу їх потенційних наслідків.

Сучасна цифрова трансформація зумовлює не лише зміну архітектурних підходів до побудови програмних систем, а й радикальну трансформацію об'єкта тестування. Якщо у традиційних підходах об'єктом перевірки виступав відносно ізольований програмний продукт, то в умовах сучасної цифрової інфраструктури таким об'єктом стає складна, динамічна екосистема взаємопов'язаних сервісів, інтеграційних модулів, середовищ виконання та інфраструктурних компонентів [5]. Масове впровадження хмарних технологій, мікросервісної архітектури, контейнеризації, інфраструктури як коду (Infrastructure as Code), а також сервісно-орієнтованих моделей (SaaS, PaaS, FaaS) зумовлює необхідність розроблення нових підходів до організації тестових процедур, орієнтованих на перевірку розподілених і масштабованих систем.

За таких умов традиційні ітеративно-циклічні моделі тестування виявляються методологічно недостатніми для забезпечення стабільності високочастотних оновлень та масштабованості сучасних цифрових

платформ. Відповіддю на зазначені виклики стало інтегрування тестування у процеси безперервної інтеграції та постачання, у межах яких кожна зміна програмного коду супроводжується автоматизованим виконанням сукупності модульних, інтеграційних і контрактних перевірок у стандартизованих і відтворюваних середовищах. У результаті тестування трансформується у безперервний, динамічний процес, що супроводжує розробку програмного забезпечення у режимі реального часу та забезпечує стабільність функціонування систем в умовах інтенсивних циклів розгортання.

Особливої уваги в межах сучасного наукового аналізу потребує констатація того, що цифрова трансформація зумовлює зміни не лише в організаційно-процесуальних аспектах тестування, а й у його змістовому наповненні. Формування подієво-орієнтованих архітектур, поширення рішень у сфері Інтернету речей (IoT) та промислового Інтернету речей (IIoT), а також розвиток інтелектуальних транспортних і мобільних систем об'єктивно спричиняють зростання значущості експлуатаційної аналітики, засобів моніторингу продуктивності та контролю параметрів якості сервісу. За таких умов тестування дедалі більшою мірою інтегрується з реальним середовищем функціонування програмних систем, що концептуалізується у підході Testing in Production, відповідно до якого окремі процедури перевірки реалізуються безпосередньо у продуктивному середовищі за умов контрольованого впливу на користувача. Унаслідок цього тестування втрачає ознаки ізольованого етапу життєвого циклу програмного забезпечення та трансформується у складову безперервного процесу експлуатації, моніторингу й адаптивного управління системою.

Разом з окресленими трансформаційними процесами формується ще одна системна тенденція – інтелектуалізація тестування, яка зумовлює принципову зміну логіки організації тестових процедур. Інтеграція методів машинного навчання у процеси пріоритизації тестових сценаріїв, прогнозування дефектів і виявлення аномалій у журналах подій забезпечує перехід від детермінованих моделей тестування до адаптивних, аналітично керованих підходів. Застосування відповідних моделей дозволяє оптимізувати тривалість виконання CI/CD-конвеєрів без зниження рівня контролю над критично значущими функціональними компонентами, а також забезпечує раннє виявлення нетипових поведінкових патернів програмних систем ще до їх трансформації у повномасштабні інциденти. У результаті тестування набуває динамічного, контекстно-орієнтованого характеру, що передбачає орієнтацію не на заздалегідь фіксований набір сценаріїв, а на актуальні ризики та емпірично зафіксовані експлуатаційні патерни.

Зазначені трансформації зумовлюють формування принципово нового підходу до інтерпретації тестованості програмного забезпечення. Якщо в межах традиційних підходів тестованість асоціювалася переважно з архітектурною прозорістю системи та можливістю формалізації тестових сценаріїв, то у сучасних умовах до її структури включаються такі характеристики, як спостережуваність, аналітична керованість та здатність системи до самодіагностики. Відповідно, тестованість постає як комплексна інтегративна властивість, що формується не лише на рівні програмного коду та архітектурних рішень, а й визначається рівнем зрілості цифрової інфраструктури, ефективністю організаційних процесів і ступенем інтеграції тестування у загальну систему управління життєвим циклом програмного забезпечення.

Отже, сучасний стан розвитку тестування програмного забезпечення засвідчує його еволюцію від локалізованої інженерної практики до системного механізму забезпечення якості, надійності, безпечності та стійкості функціонування цифрових систем. Сукупність теоретичного уточнення сутності тестованості, аналізу функціонального призначення тестування у структурі управління якістю, а також дослідження впливу цифрової трансформації на інженерні практики формує цілісну концептуально-методологічну основу для подальшого моделювання процесів тестування та розроблення формалізованих методів оцінювання якості програмного забезпечення

Список літератури

1.S. Najihi, S. Elhadi, R. Ait Abdelouahid, та A. Marzak, "Software testing from an agile and traditional view", Procedia Computer Science, т. 203, с. 775–782, 2022. DOI: https://doi.org/10.1016/j.procs.2022.07.116.

2.A. Salahirad, A. Bagheri, V. Garousi, та M. Felderer, "Mapping the structure and evolution of software testing research over the past three decades: A bibliometric assessment", Journal of Systems and Software, т. 200, ст. 111692, 2023. DOI: https://doi.org/10.1016/j.jss.2023.111692.

3.V. Garousi, M. Felderer, та F. N. Kılıçaslan, "A survey on software testability", Information and Software Technology, т. 108, с. 35–64, 2019. DOI: https://doi.org/10.1016/j.infsof.2018.12.003.

4.I. Hooda, та R. Singh Chhillar, "Software test process, testing types and techniques", International Journal of Computer Applications, т. 111, № 13, с. 10–14, 2015. DOI: https://doi.org/10.5120/19597-1433.

5.D. Sokolowski, D. Spielmann, та G. Salvaneschi, "Automated infrastructure as code program testing", IEEE Transactions on Software Engineering, опубліковано онлайн, 2024. DOI: https://doi.org/10.1109/TSE.2024.3393070.

УДК 004.421.5

О.О. Майданик, *аспірант*
А.М. Мацуй, *докт. техн. наук, професор*
Є.В. Мелешко, *докт. техн. наук, професор*
maidanyksmail@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

ПРАКТИЧНЕ ДОСЛІДЖЕННЯ МОЖЛИВОСТІ СИНХРОННОЇ ГЕНЕРАЦІЇ КЛЮЧОВОЇ ПОСЛІДОВНОСТІ НА ОСНОВІ ГПВЧ З ВИКОРИСТАННЯМ БІЛЬЯРДА СІНАЯ НА ДВОХ МІКРОКОНТРОЛЕРАХ ESP32

На даний час актуальною проблемою бездротових систем передачі даних є перехоплення керування, аналіз радіотрафіку та заглушення каналів зв'язку. Особливо це стосується систем керування безпілотними літальними апаратами (БПЛА), де недостатній рівень захисту телеметрії та командного каналу може призвести до втрати обладнання або корисної інформації. Одним із напрямів підвищення стійкості таких систем є використання шифрування з динамічною генерацією ключів без їх безпосередньої передачі каналом зв'язку [1, 2].

У попередніх дослідженнях було розглянуто генератор псевдовипадкових чисел (ГПВЧ) на основі більярда Сіная, робота якого базується на математичному моделюванні руху точки в обмеженому полі з пружним відбиттям від меж. Псевдовипадкові числа формуються на основі координат точок відбиття, а сам підхід показав добрі результати за статистичними тестами розподілу, автокореляції та періодичності [1-3].

Метою даної роботи є практичне підтвердження можливості синхронної генерації однакової псевдовипадкової послідовності на двох фізично окремих мікроконтролерах без передавання самих чисел ключа по каналу зв'язку.

Для експериментального дослідження було розроблено програмно-апаратний стенд на базі двох мікроконтролерів ESP32. Перший контролер виконував функцію передавача, другий – приймача. Передавач був підключений до персонального комп'ютера через послідовний інтерфейс UART, з якого надходили текстові повідомлення. Приймач був підключений до OLED-дисплея для відображення розшифрованого тексту.

На відміну від попередніх досліджень, де основна увага приділялася статистичним властивостям генератора псевдовипадкових чисел, у даній роботі перевірялася можливість практичного використання алгоритму в задачі захищеної передачі повідомлень між двома окремими мікроконтролерами. При цьому ключова послідовність не передавалася каналом зв'язку, а генерувалася незалежно на обох пристроях за однаковим алгоритмом. На рис. 1 наведено структурну схему експериментального стенда.

Рису. 1. Структурна схема експериментального стенда

Перед початком передачі, обидва мікроконтролери ініціалізують модулі зв'язку, службові змінні та параметри генератора псевдовипадкових чисел на основі більярда Сіная. Після цього передавальний модуль очікує надходження повідомлення з персонального комп'ютера. В експерименті використовувалося текстове повідомлення фіксованої довжини 10 символів, що спрощує обробку пакета, шифрування та виведення на дисплей приймального пристрою.

Після отримання повідомлення передавач запускає процедуру шифрування. Для кожного символу повідомлення формується черговий байт ключової послідовності, згенерований на основі алгоритму більярда Сіная. Далі виконується операція XOR між байтом відкритого тексту та згенерованим ключовим байтом. У результаті утворюється масив зашифрованих даних, який передається через бездротовий канал Wi-Fi на приймальний контролер.

Особливістю реалізованого алгоритму є необхідність синхронізації генератора на обох пристроях. Оскільки ключова послідовність не передається, приймальний контролер повинен точно знати, з якого елемента псевдовипадкової послідовності слід почати розшифрування конкретного пакета. Для цього разом із даними

використовується службова інформація про позицію або номер кроку генератора. Це дає змогу обом мікроконтролерам синхронно формувати однакові ключові байти без обміну самими значеннями ключа.

На рис.2 наведено блок-схему передачі даних.

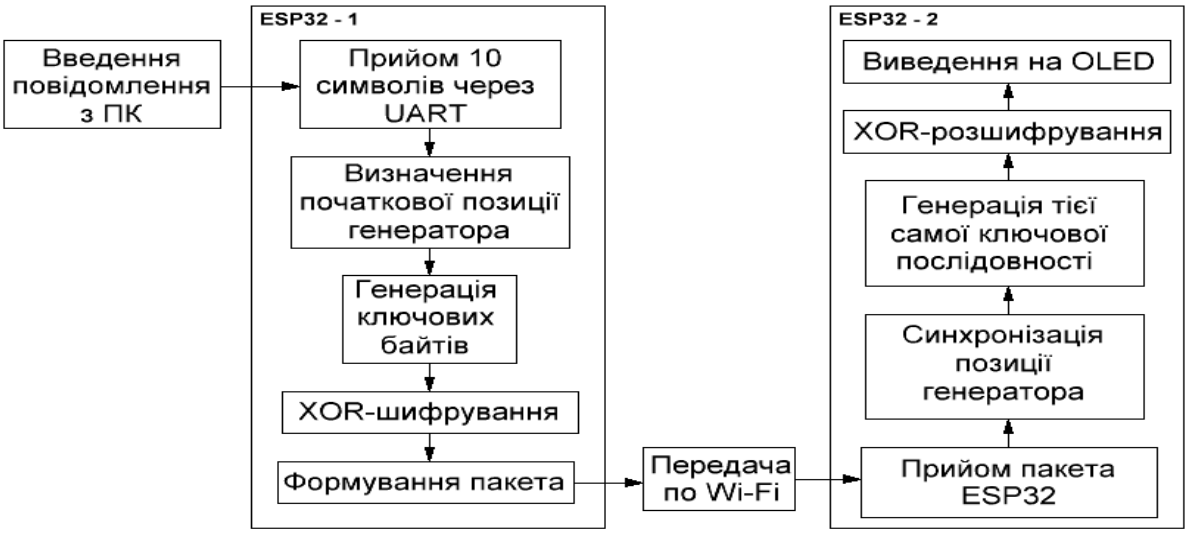


Рис. 2. Блок схема передачі даних

Алгоритм роботи передавального контролера складається з таких етапів:

1.ініціалізація UART та бездротового інтерфейсу;

2.прийом текстового повідомлення з ПК;

3.обмеження або формування повідомлення фіксованою довжиною 10 символів;

4.визначення поточного індексу генератора псевдовипадкової послідовності;

5.генерація 10 ключових байтів;

6.XOR-шифрування кожного символу повідомлення;

7.формування пакета даних, який містить службову інформацію для синхронізації та зашифроване повідомлення;

8.передача пакета по Wi-Fi.

Алгоритм роботи приймального контролера складається з таких етапів:

1.ініціалізація Wi-Fi та OLED-дисплея;

2.прийом пакета від передавача;

3.зчитування службової інформації про початкову позицію генератора;

4.генерація необхідної кількості ключових байтів у тій самій послідовності;

5.XOR-розшифрування прийнятих даних;

6.формування текстового повідомлення;

7.виведення розшифрованого тексту на OLED-дисплей.

Таким чином, у ході практичного експерименту було підтверджено, що два окремі мікроконтролери ESP32 можуть синхронно генерувати однакову псевдовипадкову послідовність і використовувати її для шифрування та розшифрування повідомлень без передавання самих ключових значень каналом зв'язку. Це підтверджує можливість практичного застосування алгоритму ГПВЧ на основі більярда Сіная у системах захищеного бездротового обміну даними.

Список літератури

1. Sinai Y.G. Dynamical systems with elastic reflections // Mathematical Surveys. – 1970. - vol. 25, no. 2, pp. 137-189.

2. Інформаційна безпека та комп'ютерні технології : VI Міжнар. наук.-практ. конф., 20-21 трав. 2023 р., м. Кропивницький : тези доп. / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програмного забезпечення. - Кропивницький : ЦНТУ, 2023. С 39 – 40.

3. Комплексне забезпечення якості технологічних процесів та систем XV Міжнародна науково-практична конференція м. Чернігів, Національний університет "Чернігівська політехніка". Том 2. С 310 – 311.

УДК 004.41:004.8

Б.О. Тарасенко[1], А.С. Коваленко[1]
tarasenko.b.o.777@gmail.com, annasun911@gmail.com
[1]Центральноукраїнський національний технічний університет, Кропивницький

ОГЛЯД СУЧАСНИХ ПІДХОДІВ ДО ОПТИМІЗАЦІЇ ІТЕРАЦІЙНОГО ПЛАНУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З УРАХУВАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ОБМЕЖЕНЬ

Сучасне розроблення програмного забезпечення характеризується високою динамікою змін, скороченням життєвих циклів програмних продуктів, потребою в регулярному уточненні вимог та необхідністю оперативного узгодження технічних і бізнесових пріоритетів. У таких умовах особливого значення набуває ітераційне планування, яке визначає зміст чергової ітерації, розподіл задач між виконавцями, очікуваний обсяг робіт, рівень ризику та досяжність цілей розроблення. У Scrum планування спринту формує основу Sprint Backlog, який поєднує ціль спринту, вибрані елементи Product Backlog та план їх виконання.

Проблема ітераційного планування полягає в тому, що вибір задач для реалізації не є лінійною процедурою. Команда одночасно враховує пріоритетність вимог, трудомісткість, залежності між задачами, доступність фахівців, рівень компетентності виконавців, очікувану бізнесову цінність, ризики невиконання, вплив технічного боргу та обмеження за часом. Тому планування ітерацій доцільно розглядати як багатокритеріальну задачу оптимізації, у якій критерії якості плану можуть конфліктувати між собою, а процес прийняття рішень додатково обмежується часовими, ресурсними, компетентнісними та технічними умовами.

У цьому контексті доцільним є розгляд сучасних наукових праць, у яких ітераційне планування програмного забезпечення досліджується як задача оптимізації, інтелектуальної підтримки прийняття рішень і адаптивного розподілу ресурсів в умовах багатокритеріальних обмежень. У дослідженні X. N. Shen, L. L. Minku, N. Marturi та співавторів [1] запропоновано алгоритм на основі Q-навчання і меметичного пошуку для динамічного планування програмних проєктів, що підтверджує можливість поєднання підкріплювального навчання з багатокритеріальною оптимізацією тривалості, вартості, стійкості плану та задоволеності учасників команди. Праця C. F. Hayes, R. Radulescu, E. Bargiacchi та співавторів [2] обґрунтовує доцільність використання багатоцільового підкріплювального навчання для задач прийняття рішень, у яких необхідно одночасно враховувати кілька взаємно конфліктних критеріїв. У роботі H. K. Dam, T. Tran, J. Grundy та співавторів [3] показано, що засоби штучного інтелекту можуть підтримувати гнучке управління проєктами через прогнозування, оцінювання ризиків, рекомендаційні механізми та підвищення обґрунтованості рішень під час планування спринтів. Огляд D. Wang, H. You, L. Zhu та співавторів [4] засвідчує активне формування напряму застосування підкріплювального навчання в інженерії програмного забезпечення, зокрема для автоматизації, оптимізації, тестування, супроводу та підтримки прийняття рішень у програмних системах. Дослідження J. Shi, H. Lou, X. Shen та співавторів [5] розглядає планування гнучких проєктів для кількох команд як задачу оптимізації, у якій враховуються вибір user stories, призначення виконавців, досвід учасників, доступність ресурсів та зміни в процесі розроблення.

Вітчизняні дослідження також підтверджують актуальність інтелектуалізації процесів планування та розподілу ресурсів в IT-проєктах. У праці O. Yanholenko, M. Grinchenko, M. Rohovyi та співавторів [6] запропоновано модель інтелектуального планування командної роботи, орієнтовану на підвищення цінності спринту через урахування характеристик задач, переваг учасників команди, бізнесової цінності та ризиків. Дослідження O. Kuchanskyi, M. Gladka, Y. Hladkyi та співавторів [7] акцентує увагу на алгоритмічному розподілі трудових ресурсів в IT проєктах з урахуванням компетентностей, завантаження, вартості та строків виконання робіт. У роботі V. Druzhynin, M. Gladka, I. Borysenko та співавторів [8] запропоновано модель розподілу трудових ресурсів на основі пріоритезації задач, що є важливим для ітераційних моделей розроблення, де склад задач спринту постійно змінюється і потребує оперативного коригування.

Аналіз наведених публікацій дає змогу виокремити кілька ключових тенденцій розвитку цього напряму.

1. Сучасні підходи підтверджують перехід від переважно експертного формування плану ітерації до моделей, які формалізують вибір елементів беклогу, їх пріоритезацію, розподіл між виконавцями, урахування залежностей, прогнозування ризиків і контроль відповідності плану доступним ресурсам.

2. Важливою ознакою сучасних підходів є багатокритеріальний характер планування. Вибір задач для ітерації не може ґрунтуватися лише на одному показнику, оскільки програмні проєкти потребують одночасного врахування строків, вартості, якості, завантаження команди, технічної складності, бізнесової цінності, ризиків невиконання та впливу поточних рішень на наступні ітерації. Тому використання багатокритеріальних моделей створює передумови для більш збалансованого й обґрунтованого формування плану розроблення.

3. Окремого значення набуває адаптивність планування. Статичні моделі, що використовують наперед задані вагові коефіцієнти або фіксовані правила вибору задач, не завжди ефективно працюють у динамічному середовищі, де змінюються вимоги, доступність фахівців, рівень ризику, фактична продуктивність, стан беклогу та технічні обмеження. За таких умов перспективним є застосування підкріплювального навчання, яке

дає змогу формувати політику прийняття рішень на основі взаємодії із середовищем і результатів попередніх ітерацій.

У межах адаптивного підходу ітераційне планування доцільно подати як процес послідовного прийняття рішень, у межах якого стан системи містить дані про беклог, пріоритети вимог, залежності між задачами, компетентності виконавців, часові ресурси, історичну продуктивність команди, ризики, технічний борг і результати попередніх ітерацій. Діями агента можуть бути вибір задач до ітерації, зміна їхньої пріоритетності, розподіл між виконавцями або коригування плану, а вектор винагород або агрегована функція винагороди може враховувати виконання плану, бізнесову цінність, зниження ризику, раціональне використання ресурсів, збереження якості та мінімізацію негативного впливу технічного боргу.

Проведений огляд показує, що сучасні підходи до ітераційного планування програмного забезпечення розвиваються в напрямі інтелектуалізації, багатокритеріальної оптимізації та адаптивного прийняття рішень. Зарубіжні дослідження підтверджують перспективність використання підкріплювального навчання і багатоцільових моделей для динамічного планування програмних проєктів, а українські праці демонструють актуальність задач інтелектуального планування спринтів, пріоритезації задач і розподілу трудових ресурсів в IT-проєктах. Водночас наявні підходи здебільшого розглядають окремі складові проблеми: загальне планування програмних проєктів, гнучке управління проєктами, розподіл трудових ресурсів або теоретичні засади багатоцільового підкріплювального навчання. Пряме поєднання ітераційного планування, багатокритеріальних обмежень і підкріплювального навчання залишається недостатньо розвиненим, тому науково й практично актуальним є дослідження, спрямоване на розроблення інформаційної технології оптимізації ітераційного планування програмного забезпечення з урахуванням багатокритеріальних обмежень із застосуванням підкріплювального навчання.

Список літератури

1. Shen X.-N., Minku L. L., Marturi N. et al. A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling. Information Sciences. 2018. Vol. 428. P. 1–29. URL: https://doi.org/10.1016/j.ins.2017.10.041; https://minkull.github.io/publications/ShenINF2017.pdf (дата звернення: 19.04.2026).

2. Hayes C. F., Radulescu R., Bargiacchi E. et al. A practical guide to multi-objective reinforcement learning and planning. Autonomous Agents and Multi-Agent Systems. 2022. Vol. 36. Art. 26. URL: https://doi.org/10.1007/s10458-022-09552-y (дата звернення: 19.04.2026).

3. Dam H. K., Tran T., Grundy J. et al. Towards effective AI-powered agile project management. 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). 2019. P. 41–44. URL: https://doi.org/10.1109/ICSE-NIER.2019.00019 https://arxiv.org/pdf/1812.10578 (дата звернення: 19.04.2026).

4. Wang D., You H., Zhu L. et al. A Survey of Reinforcement Learning for Software Engineering. arXiv. 2025. URL: https://doi.org/10.48550/arXiv.2507.12483 (дата звернення: 19.04.2026).

5. Shi J., Lou H., Shen X. et al. Multi-Team Agile Software Project Scheduling Using Dual-Indicator Group Learning Particle Swarm Optimization. Symmetry. 2025. Vol. 17, no. 8. Art. 1267. URL: https://doi.org/10.3390/sym17081267 (дата звернення: 19.04.2026).

6. Yanholenko O., Grinchenko M., Rohovyi M. et al. Intelligent team work planning: a model for increasing sprint value. CEUR Workshop Proceedings. 2025. Vol. 4015. P. 134–149. URL: https://ceur-ws.org/Vol-4015/paper10.pdf (дата звернення: 19.04.2026).

7. Kuchanskyi O., Gladka M., Hladkyi Y. et al. The algorithm for labor resource allocation in IT project implementation based on stochastic gradient descent. CEUR Workshop Proceedings. 2025. Vol. 4160. P. 246–257. URL: https://ceur-ws.org/Vol-4160/paper13.pdf (дата звернення: 19.04.2026).

8. Druzhynin V., Gladka M., Borysenko I. et al. A Model for Allocating Labor Resources to Project Work Based on Task Prioritization. CEUR Workshop Proceedings. 2024. Vol. 3955. P. 42–54. URL: https://ceur-ws.org/Vol-3955/Paper_4.pdf (дата звернення: 19.04.2026).

СЕКЦІЯ 3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ, МЕДИЦИНІ ТА ОСВІТІ

УДК 37.016:811.1/.2]:004.9

Н.В. Кіш, О.Л. Канюк
nadiia.kish@uzhnu.edu.ua, oleksandra.kanyuk@uzhnu.edu.ua
ДВНЗ »Ужгородський національний університет», Ужгород

ЦИФРОВЕ СЕРЕДОВИЩЕ ЯК ДРАЙВЕР АКАДЕМІЧНОЇ ТРАНСФОРМАЦІЇ У ВИВЧЕННІ ІНОЗЕМНИХ МОВ

Сучасна парадигма вищої освіти перебуває у стані фундаментальної трансформації, зумовленої стрімким розвитком глобального цифрового простору. Традиційна модель «трансляції знань», де викладач виступає ретранслятором інформації, а студент — пасивним реципієнтом, вичерпала свій ресурс і не відповідає запитам сучасного суспільства.

Сьогодні освітній вектор зміщується до створення динамічних, персоналізованих цифрових екосистем, що базуються на засадах активної взаємодії, автономії суб'єктів навчання та безперервної інтеграції інноваційних технологій у професійний контекст.

Особливої гостроти ця проблема набуває у сфері іншомовної підготовки. В умовах високотехнологічного середовища іноземна мова перестає бути лише об'єктом вивчення, трансформуючись у критично важливий інструмент фахової комунікації. Проте спостерігається суттєве протиріччя між зростаючими вимогами ринку праці до цифрової та комунікативної мобільності фахівців і застарілими репродуктивними методами навчання, що все ще домінують у вищій школі.

Виникає нагальна потреба у розбудові такої моделі навчання, де використання штучного інтелекту, мобільних платформ та імерсивних технологій не буде фрагментарним доповненням, а стане технологічним ядром формування професійної компетентності. Цифровізація освіти в цьому контексті виступає не просто технічним оновленням засобів навчання, а стратегічним напрямом реконцептуалізації всієї системи підготовки, що стимулює перехід від класичних університетських структур до відкритих цифрових екосистем.

Цифрове середовище стає ключовим фактором трансформації сучасної академічної освіти, забезпечуючи нові можливості для навчання, досліджень та розвитку професійних компетентностей студентів. Воно не лише підтримує традиційні методи навчання, а й створює інноваційні форми взаємодії між студентами та викладачами, сприяючи персоналізації освітнього процесу та активному залученню студентів. Це потребує глибокого аналізу оновленої структури компетентностей фахівця та нових підходів до організації педагогічної взаємодії у віртуальному просторі, що дозволить студентам ефективно інтегруватися у глобальне професійне середовище.

Аналіз останніх досліджень і публікацій свідчить [2], що цей процес безпосередньо впливає на методичні підходи до навчання іноземних мов, вимагаючи відходу від класичних репродуктивних моделей. Поява дистанційного навчання, інтерактивних платформ, штучного інтелекту (ШІ) та мобільних технологій відкриває нові перспективи для формування професійної комунікативної компетентності майбутніх фахівців.

Як зазначають дослідники, цифрові технології в освітньому процесі сприяють не лише розвитку мовних навичок, але й формуванню складних когнітивних та професійних компетенцій. Це є критично важливим у контексті вимог сучасного глобалізованого ринку праці, де іноземна мова виступає не просто предметом вивчення, а інструментом фахової діяльності. На думку Бобро Н. В. [1], цифровізація освіти стимулює трансформацію університетських моделей у відкриті цифрові екосистеми, що створює сприятливий фундамент для неперервного професійного зростання.

Платформи дистанційного та змішаного навчання

Сучасна академічна освіта все більше інтегрує дистанційні та змішані форми навчання, що дозволяють поєднувати очний та онлайн-формат навчання. Ключовим елементом таких підходів є цифрові платформи, які забезпечують організацію навчального процесу, взаємодію студентів і викладачів, доступ до навчальних матеріалів і контроль результатів.

Онлайн-середовища (LMS: Moodle, Canvas), відеоконференції (Zoom, Teams) та інтерактивні сервіси (Mentimeter, Quizlet) забезпечують гібридне навчання, що поєднує самостійну роботу та комунікацію у групах.

Таблиця 1.

Порівняння традиційного та цифрового підходу до навчання

Параметр	Традиційне навчання	Цифрове навчання
Доступність	Лекції та аудиторні заняття	24/7 доступ до онлайн-курсів, ресурсів
Інтерактивність	Обмежена	Взаємодія через форуми, відеоконференції, VR/AR
Персоналізація	Мінімальна	Адаптивні курси, AI-підтримка, мікрокреденціали
Моніторинг прогресу	Ручний контроль	Цифрові портфоліо, автоматизовані тести
Мотивація	Підпорядкована викладачу	Гейміфікація, нагороди, онлайн-змагання

Розглянемо ключові відмінності між традиційним та цифровим навчанням, відображаючи зміни, які відбуваються в освітньому середовищі в умовах цифрової трансформації.

Доступність

У традиційному навчанні доступ до матеріалів обмежений фізичним місцем і розкладом (лекції, аудиторні заняття). Цифрове навчання дає можливість користуватися онлайн-курсами і ресурсами цілодобово (24/7), що забезпечує гнучкість і можливість навчатися з будь-якого місця. Цей аспект підсилює право на освіту, особливо для тих, хто має обмежений час, географічні або фізичні обмеження.

Інтерактивність

Традиційне навчання обмежено взаємодією: переважно лекції з питаннями-відповідями. У цифровому навчанні взаємодія значно розширюється: онлайн-форуми, відеоконференції, віртуальна (VR) та доповнена реальність (AR) . Такі технології створюють більш «іммерсивне» (занурююче) середовище, яке може покращувати залучення та ефективність навчання.

Крім того, дослідження показують, що цифрові платформи освітнього процесу можуть значно підвищувати інтерактивність завдяки мультимедійним компонентам, симуляціям, іграм тощо.

Персоналізація

У традиційній освіті часто застосовується одна програма для всіх студентів. Цифрове навчання дозволяє створювати адаптивні курси: системи адаптивного навчання (adaptive learning), хмарні рішення, персоналізація контенту та оцінювання. Також дослідження показують, що адаптивні технології у поєднанні з AR можуть краще враховувати індивідуальні потреби студентів (наприклад, з особливими освітніми потребами).

Моніторинг прогрессу

У традиційному навчанні оцінювання й контроль часто залежать від викладача: перевірка робіт, індивідуальні консультації, класні тести. Цифрове навчання дає змогу автоматизувати ці процеси через онлайн-тести, цифрові портфоліо, LMS-системи, і надавати зворотний зв'язок в реальному часі. Також у контексті цифрових освітніх платформ дослідники вказують, що саме за допомогою таких платформ можна більш гнучко й точно відстежувати прогрес і коригувати навчальні траєкторії.

Мотивація

У традиційній системі мотивація студентів часто залежить від викладача, оцінок і класних активностей. У цифровому навчанні поширені такі механізми, як гейміфікація, системи нагород і змагання, які підвищують мотивацію й самостійність студентів. Крім того, дослідження педагогічних підходів показують, що конструктивістські, проблемно-орієнтовані й особистісно-орієнтовані методи, у поєднанні з VR, AR і адаптивними системами, значно підвищують залучення та мотивацію під час дистанційного навчання.

Порівняльний аналіз традиційного та цифрового підходів до навчання засвідчує суттєву трансформацію освітнього процесу в умовах цифровізації.

Таким чином, платформи дистанційного та змішаного навчання відіграють ключову роль у сучасній професійній іншомовній освіті, забезпечуючи інтеграцію цифрових технологій, гнучкість навчального процесу та персоналізацію навчання. Вони створюють умови для ефективного доступу до освітніх ресурсів, активної комунікації та колаборації студентів, а також підтримують системне оцінювання і моніторинг прогресу.

Завдяки таким платформам студенти отримують можливість застосовувати мовні знання у практичних професійних ситуаціях, розвивати цифрову та інформаційну компетентність, а викладачі – оптимізувати процес організації навчання і контролю його результатів.

Отже, використання платформ дистанційного та змішаного навчання сприяє формуванню інтегрованої іншомовної та професійної компетентності, готової до вимог сучасного академічного та професійного середовища.

Мобільне навчання, AI та адаптивні системи

Сучасна професійна іншомовна освіта вимагає інтеграції традиційних методів навчання з цифровими технологіями, що забезпечують більш ефективне засвоєння знань та розвиток ключових компетентностей.

Використання цифрових інструментів у навчальному процесі дозволяє створити середовище, спрямоване на розвиток академічних, професійних та комунікативних умінь студентів, формуючи навички, необхідні для успішної діяльності в міжнародному контексті.

Мобільні додатки дозволяють студентам вивчати мову у зручному темпі й у будь-яких життєвих ситуаціях, розширюючи можливості позааудиторного навчання.

Адаптивні AI-системи забезпечують індивідуальне налаштування навчального маршруту: аналізують прогрес, складність помилок, швидкість засвоєння матеріалу, пропонують персоналізовані вправи й надають миттєвий зворотний зв'язок, що значно підвищує ефективність оволодіння мовою.

Мікрокреденціали (micro credentials) стають важливим інструментом сертифікації вузьких професійних компетенцій та відіграють зростаючу роль у системі професійної сертифікації, надаючи студентам можливість швидше адаптуватися до вимог міжнародного ринку праці.

Рис. 1. Цифрові інструменти у професійній іншомовній освіті

Отже, комплексна цифровізація іншомовної освіти є безальтернативним шляхом розвитку сучасної вищої школи, оскільки трансформує структуру професійної компетентності фахівця, інтегруючи лінгвістичний базис із навичками управління даними та культурою віртуальної взаємодії. Цей процес зумовлює реконцептуалізацію ролі педагога як цифрового фасилітатора, який моделює персоналізований освітній досвід за допомогою синергії AI, VR/AR та моделі «перевернутого класу». Створення багаторівневої цифрової екосистеми на базі LMS забезпечує прозорість моніторингу результатів та стає стратегічним каталізатором професійного зростання майбутнього спеціаліста.

Таким чином, мобільне навчання, штучний інтелект та адаптивні освітні системи стають ключовими драйверами модернізації професійної іншомовної підготовки. Поєднання мобільного навчання, штучного інтелекту та мікрокреденціалів формує сучасну цифрову освітню екосистему, що сприяє розвитку іншомовної професійної комунікативної компетентності, підвищує автономію студентів та забезпечує їхню готовність до професійної діяльності у глобальному середовищі.

Список літератури

1.Бобро Н.В. Цифровізація освіти в контексті формування університету нового покоління. Український педагогічний журнал. 2025. № 2. С. 27–34.

2.Кіш Н. В., Канюк О. Л. Цифровізація іншомовної освіти як стратегічний напрям професійної підготовки майбутніх фахівців. *Вісник науки та освіти (Серія «Філологія», Серія «Педагогіка», Серія «Соціологія», Серія «Культура і мистецтво», Серія «Історія та археологія»)*: журнал. 2026. № 2 (44). С. 730-733.

УДК 004.052

Т.Х. Фаталієв
tfataliyev@gmail.com
Інститут інформаційних технологій, м. Баку, Азербайджан

ЗАСТОСУВАННЯ ФЕДЕРАТИВНОГО НАВЧАННЯ ПІДВИЩЕННЯ НАДІЙНОСТІ IOT-ОРІЄНТОВАНИХ КІБЕРФІЗИЧНИХ СИСТЕМ НАФТОГАЗОВОГО КОМПЛЕКСУ

Сучасний нафтогазовий комплекс (НГК) характеризується високим ступенем цифровізації технологічних процесів видобутку, транспортування та переробки вуглеводневої сировини. Кіберфізичні системи, побудовані на основі розподілених сенсорних мереж та промислових платформ Інтернету речей (IIoT), забезпечують безперервний моніторинг параметрів свердловин, магістральних трубопроводів, компресорних станцій, резервуарних парків та установок глибокої переробки. При цьому надійність таких систем стає ключовим фактором промислової безпеки, стійкості виробничих циклів та мінімізації екологічних ризиків. В умовах територіальної розподіленості об'єктів, обмежених енергетичних ресурсів та підвищеної кіберзагроз традиційні централізовані підходи до обробки даних та навчання моделей штучного інтелекту демонструють обмеження, пов'язані із затримками передачі, вразливістю до атак та ризиком каскадних відмов.

Традиційні централізовані моделі машинного навчання припускають передачу великих масивів даних у центр обробки. Однак НГК така модель збільшує навантаження на канали зв'язку, створює єдині точки відмови та підвищує ризики порушення інформаційної безпеки. Компрометація центрального вузла чи втрата зв'язку можуть призвести до деградації інтелектуальних функцій діагностики та прогнозування, що безпосередньо впливає надійність технологічних процесів.

Федеративне навчання (ФН) є розподіленим підходом до побудови інтелектуальних моделей, у якому дані залишаються на локальних об'єктах, але в центральний сервер передаються лише оновлені параметри моделей. Такий механізм дозволяє поєднувати переваги колективного навчання із збереженням автономності кожного промислового вузла. Для НГК це має важливе значення, оскільки об'єкти часто територіально розподілені і функціонують за умов обмеженої пропускної спроможності каналів зв'язку. Так, ФН розглядається як перспективний механізм розподіленої інтелектуальної обробки даних, що дозволяє навчати моделі без передачі вихідних масивів інформації в центральне сховище. Архітектурні рішення, орієнтовані на IIoT-середовище, передбачають використання edge-обчислень та динамічного вибору пристроїв для участі у навчанні, що знижує навантаження на канали зв'язку та зменшує енергоспоживання [1]. Для НГК це особливо актуально при експлуатації віддалених родовищ, морських платформ та важкодоступних ділянок трубопроводів, де стабільність каналів зв'язку обмежена, а відмова центрального сервера може призвести до втрати керованості технологічним процесом.

З погляду безвідмовності застосування федеративного навчання сприяє зниженню ймовірності системних збоїв. Локальні моделі продовжують функціонувати навіть за тимчасової недоступності центрального сервера, що агрегує. Таким чином, інтелектуальні функції діагностики обладнання, виявлення аномалій та прогнозування відмов не припиняють роботу повністю, а переходять в автономний режим. Це підвищує можливість збереження коректного управління технологічним процесом.

З погляду критеріїв надійності ФН сприяє підвищенню стійкості до відмови за рахунок розподілу обчислювального навантаження між безліччю вузлів. При виході з ладу окремого сенсора чи промислового контролера система продовжує функціонувати, оскільки глобальна модель формується з урахуванням агрегованих оновлень від доступних пристроїв. Такий підхід зменшує ймовірність повної зупинки аналітичного контуру та знижує залежність від єдиної точки відмови. Безвідмовність функціонування зростає завдяки локальній перед обробці даних і можливості автономної адаптації моделей до змінних технологічних умов, наприклад, коливань дебіту свердловини або зміни складу суміші, що транспортується.

У завданнях забезпечення промислової безпеки особливе значення має виявлення аномалій та кібератак. Асинхронні схеми ФН з гібридними нейро мережевими моделями дозволяють виявляти вторгнення та відхилення в мережевому трафіку без необхідності централізованого збору чутливих даних [2]. Це знижує ризик компрометації інформації про критичну інфраструктуру і одночасно підвищує живучість системи, оскільки навіть за часткового порушення комунікацій локальні вузли продовжують аналізувати події та формувати попередження. У разі НГК своєчасне виявлення аномалій запобігає розвитку аварійних сценаріїв, що з перевищенням тиску, температурних режимів чи концентрації вибухонебезпечних сумішей.

Додатковий внесок у підвищення надійності робить використання адаптивних вагових коефіцієнтів при агрегуванні локальних моделей. Підходи, засновані на оцінці вкладу кожного учасника навчання, дозволяють враховувати неоднорідність даних, відмінності як сенсорну інформацію та обчислювальні можливості вузлів [3]. У НГК це важливо за інтеграції даних з об'єктів різної технологічної зрілості: від старих родовищ із застарілою автоматикою до сучасних цифрових платформ. Коректне зважування оновлень підвищує стійкість

глобальної моделі до спотворень та зменшує ймовірність помилкових спрацьовувань, що впливає на показник безвідмовності аналітичної підсистеми.

Гібридні двоетапні схеми ФН орієнтовані попереднє локальне виявлення аномалій з наступною централізованою класифікацією агрегованих ознак. У статті [4] представлена двоетапна гібридна структура ФН для виявлення та класифікації аномалій у IoT. На першому етапі кожен пристрій навчає генеративну модель II тільки на безпечному трафіку, а на другому етапі класифікатор на основі гістограмного градієнтного бустингу надає мітки поміченому трафіку. Така архітектура поєднує переваги розподіленої обробки та глобального аналізу, забезпечуючи баланс між швидкістю реакції та глибиною діагностики. Для магістральних трубопроводів це дозволяє оперативно фіксувати витоки, несанкціоновані врізання та вібраційні відхилення, а потім уточнювати тип інциденту на рівні корпоративного центру моніторингу. Підвищення точності класифікації знижує можливість розвитку каскадних відмов.

Дослідження масштабованості федеративних моделей в IoT-мережах підтверджують їх стійкість до збільшення обсягу даних та кількості підключених пристроїв. У статті [5] представлено масштабне емпіричне дослідження, спрямоване визначення оптимальної локальної моделі глибокого навчання та обсягу даних для розгортання систем виявлення вторгнень на пристроях IoT з обмеженими ресурсами з використанням ФН. Для нафтогазової інфраструктури, де число сенсорів може обчислюватися десятками тисяч, масштабованість є критичним умовою надійності. Здатність системи адаптуватися до зростання інформаційного потоку без деградації якості прогнозування означає збереження високої доступності сервісів та мінімізацію простою обладнання.

Аналітичне порівняння існуючих підходів показує, що архітектури, орієнтовані виключно на зниження затримок та енергоспоживання, потребують доповнення механізмами оцінки вкладу вузлів та захисту від несумлінних учасників. Моделі, сфокусовані на кібербезпеці, демонструють високу точність виявлення атак, проте потребують інтеграції з технологічними параметрами виробничих процесів. Комплексні гібридні рішення, що поєднують асинхронне навчання, адаптивне зважування та двоетапну фільтрацію аномалій, є найбільш перспективними для критично важливої нафтогазової інфраструктури.

З позиції теорії надійності застосування ФН впливає на показники відмово стійкості, безвідмовності та живучості системи. Відмово стійкість забезпечується за рахунок відсутності єдиної точки відмови та можливості динамічного перерозподілу завдань. Безвідмовність підвищується завдяки безперервному самонавчання моделей і адаптації до експлуатаційних умов, що змінюються. Живучість проявляється у здатності системи зберігати базову функціональність навіть за часткової деградації комунікаційного середовища чи цілеспрямованих кібератаках. У сукупності це знижує інтегральний ризик технологічних порушень та сприяє запобіганню великим аваріям.

В умовах видобутку на морських платформах та в арктичних регіонах стійкість цифрових систем до зовнішніх впливів набуває особливої значущості. Обмежена пропускна спроможність супутникових каналів і висока вартість передачі роблять ФН економічно виправданим. Локальна обробка інформації зменшує затримки та підвищує оперативність прийняття рішень, що критично при управлінні проти викидним обладнанням або системами підтримки пластового тиску. Аналогічно, на етапах транспортування нафти та газу розподілені інтелектуальні вузли дозволяють швидко локалізувати інциденти та запобігти поширенню негативних ефектів по всій мережі.

Таким чином, ФН формує технологічну основу для побудови надійних, масштабованих та стійких IoT-орієнтованих кіберфізичних систем НГК. Інтеграція розподілених методів машинного навчання з промисловими стандартами безпеки сприяє зниженню ймовірності каскадних відмов, мінімізації простоїв та підвищенню рівня екологічної захищеності. Подальші дослідження доцільно направити розробку галузевих методик оцінки надійності федеративних архітектур, і навіть на експериментальну апробацію моделей за умов реальних виробничих об'єктів.

Список літератури

1. Liu X., Dong X., Jia N., Zhao W., Federated Learning-Oriented Edge Computing Framework for the IIoT [Electronic resource]. Sensors. 24(13), 2024, 4182. DOI: https://doi.org/10.3390/s24134182

2. Bukhari S., et al., Enhancing cybersecurity in Edge IIoT networks: An asynchronous federated learning approach with a deep hybrid detection model [Electronic resource]. Internet of Things, 27(9), 2024, 101252. DOI: https://doi.org/10.1016/j.iot.2024.101252

3. Bhatti D., Ali M., Yoon J., Choi B., Efficient Collaborative Learning in the Industrial IoT Using Federated Learning and Adaptive Weighting Based on Shapley Values [Electronic resource]. Sensors, 25(3), 2025, 969. DOI: https://doi.org/10.3390/s25030969.

4. Shahin M., Hosseinzadeh A., Chen F., A Two-Stage Hybrid Federated Learning Framework for Privacy-Preserving IoT Anomaly Detection and Classification [Electronic resource]. IoT, 6(3), 2025, 48. DOI: https://doi.org/10.3390/iot6030048.

5. Albanbay N., et al., Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study [Electronic resource]. Journal of Sensors and Actuator Networks, 14(4), 2025, 78, DOI: https://doi.org/10.3390/jsan14040078.

УДК 378:001.89:004

T. Zhyrova , N. Kotenko
zhyrova@outlook.com, kotenkono@knute.edu.ua
State University of Trade and Economics, Kyiv

LEVERAGING DIGITAL PRODUCTS TO DEVELOP THE RESEARCH COMPETENCE OF IT MASTER'S STUDENTS

Preparing master's students in Information Technology for a rapidly changing digital society requires more than advanced programming or systems-design skills. A defining attribute of the IT graduate is research competence, such as a composite of methodological literacy, problem formulation, ethical and legal awareness, data management, reproducibility, scholarly communication, and the ability to use digital infrastructures for inquiry. A recent survey shows that master's students often do not view the research component as important for their professional growth; most lack a clear understanding of how to conduct research properly and which tools can support the process.

Research competence in higher education increasingly intersects with digital competence: the ability to find, critically appraise, and apply evidence; manage data and code; ensure reproducibility; and communicate results openly and verifiably. Systematic reviews confirm that widely used frameworks (e.g., DigComp, DigCompEdu, TPACK) help operationalize these abilities; however, measurement quality still depends on institutional context and staff preparedness such as hence the need for tightly scoped programmatic choices and clear assessment procedures rather than broad "tool catalogues" [4, 9].

Validated instruments exist for assessing students' digital competence and can be adapted to IT master's programmes. These instruments emphasise technical, informational, and ethical dimensions such as directly aligned with our minimal stack (literature search, reference management, project registration, version control, analysis, identity/dissemination, governed AI) and enable before/after evaluation of curriculum changes aimed at strengthening research competence [6].

Within computing curricula, reproducibility has become a key learning outcome grounded in version control, transparent data pipelines, and testable code. Position papers and teaching cases argue for embedding reproducibility "about, with, and for" computing (git-centred workflows, notebooks, automated checks) from early coursework through the thesis stage; even first-year projects can be structured as reproducible studies when supported by coherent tooling and rubrics [1, 2].

Artificial intelligence in education brings both opportunities and risks. Recent reviews document efficiency gains and research support while stressing governance: source transparency, academic integrity, and data protection. Policy guidance converges on human-in-the-loop use, explicit safeguards, and programme-level consistency aligning with our single-stack principle (institutional alignment to either a Microsoft or a Google ecosystem) to reduce friction and sustain student motivation [5, 7, 8]. Student surveys indicate that AI is already widely used, yet learners expect clear guidance on proper use and verification which supports our requirement to pair AI-assisted search/coding with source attribution and tests (unit tests for code; reproducible notebooks for analyses) [3].

To avoid cognitive overload and loss of motivation, research training for IT master's students should be organised around a minimum viable toolchain rather than extensive catalogues of options. Instructors must balance what can be offered with what should be required, adding tools only when each one supports a specific research capability. We adhere to the principle of single-stack coherence: if a university predominantly uses Microsoft services, the stack should anchor in that ecosystem; if it uses Google tools, it should be built around them. Such alignment reduces friction, simplifies support, and sustains student engagement.

The selection of digital products for developing research competence should therefore be based not on popularity or market trends, but on pedagogical and methodological relevance. Each tool included in the learning environment should correspond to a clearly defined research task: discovering and evaluating sources, organising references, planning and documenting research procedures, managing code and data, validating results, and disseminating outputs in accordance with open-science principles. Such a criterion-based approach makes the toolchain more transparent for students and instructors and helps transform digital products from auxiliary utilities into structured instruments of research training.

Core capabilities and deliberately minimal tools:

- literature discovery: Google Scholar, arXiv;
- reference management: Zotero, Better BibTeX;
- project planning / preregistration: Open Science Framework;
- version control and collaboration: Git, GitHub or GitLab;
- analysis and validation: Python (NumPy, pandas, scikit-learn), pytest;
- author identification and dissemination: ORCID, Zenodo (DOI for data/code);
- governed AI assistance: LLM assistants for search/coding with mandatory source verification and tests.

Reference implementations of a single:

Microsoft vector: Literature (Google Scholar, arXiv) → Zotero → OSF (linked) → GitHub + VS Code → Python + pytest → ORCID → Zenodo (DOI). Collaboration and delivery via Teams/OneDrive/SharePoint; GitHub Actions for CI (linting, tests, PDF builds).

Google vector: Literature (Google Scholar, arXiv) → Zotero → OSF (linked) → Git + GitLab/GitHub → Colab (for initial analysis) or local Python + pytest → ORCID → Zenodo (DOI). Collaboration and delivery via Drive/Docs/Sheets/Meet; Colab Pro as needed.

AI assistants can accelerate search and coding, but their use must be constrained by source attribution (links to real publications) and verification (unit tests/pytest for code; reproducible notebooks for analyses). These safeguards should be made explicit in grading rubrics so that AI augments rather than replaces scientific reasoning.

The proposed minimal, coherent stack:

1. Maintains focus and lowers the entry barrier for IT master's students by limiting tools to those that are functionally necessary.
2. Reduces operational friction and support costs through a single vector (Microsoft or Google) across courses and projects.
3. Supports the full research lifecycle from discovery and study design to analysis, verification, and open dissemination (DOI/ORCID).
4. Improves transparency and reproducibility via required version control, testing, and data documentation.
5. Sustains student motivation, as each tool has a clear role in the competence model and avoids unnecessary overload.

The effectiveness of such a minimal digital toolchain depends not only on the selection of tools themselves, but also on the way they are embedded into the educational process. Research competence develops more consistently when digital products are integrated into authentic academic tasks such as literature reviews, project proposals, data analysis assignments, reproducible coding exercises, and thesis preparation. In this context, assessment criteria should explicitly address source transparency, correctness of citation, version control practice, data and code documentation, and the justified use of AI assistance..

Conclusion. This study argues that research training for IT master's students benefits from a deliberately minimal, single-stack toolchain aligned with the university's dominant ecosystem (Microsoft or Google). Centering the curriculum on a small set of interoperable tools such as literature search (Google Scholar, arXiv), reference management (Zotero), project registration (OSF), version control (Git/GitHub or GitLab), analysis and validation (Python with NumPy, pandas, scikit-learn, pytest), identity and dissemination (ORCID, Zenodo), and governed AI assistanc reduces friction, improves supportability, and sustains student motivation. Pilot evidence indicates better focus, clearer process transparency, and higher reproducibility through mandatory versioning, testing, and data documentation, while maintaining a low entry barrier.

References

1. Bean B. L. Teaching Reproducibility to First-Year College Students: A Case Study in Introductory Data Science // Journal of Effective Teaching in Higher Education. – 2023. – Vol. 7, No. 2. – URL: https://digitalcommons.usu.edu/jete/vol7/iss2/5/

2. Fund F., et al. We Need More Reproducibility Content Across the Computer Science Curriculum. – 2023. – URL: https://dl.acm.org/doi/10.1145/3589806.3600033; https://par.nsf.gov/servlets/purl/10466268

3. Jisc. Student Perceptions of AI 2025. – 2025. – URL: https://www.jisc.ac.uk/reports/student-perceptions-of-ai-2025

4. López-Nuñez J. A., et al. A Systematic Review of Digital Competence Evaluation in Higher Education // Education Sciences. – 2024. – Vol. 14, No. 11. – P. 1181. – DOI: 10.3390/educsci14111181.

5. Mustafa M. Y., et al. A Systematic Review of Literature Reviews on AIED // Smart Learning Environments. – 2024. – Vol. 11. – P. 38. – DOI: 10.1186/s40561-024-00350-5.

6. Tzafilkou K., Mavridis I., Tzafilkou M., Economides A. A. Development and Validation of Students' Digital Competence Scale (SDiCoS) // International Journal of Educational Technology in Higher Education. – 2022. – Vol. 19. – Art. 45. – DOI: 10.1186/s41239-022-00330-0.

7. U.S. Department of Education, Office of Educational Technology. Artificial Intelligence and the Future of Teaching and Learning: Insights and Recommendations. – Washington, DC: U.S. Department of Education, 2023. – URL: https://www.ed.gov/sites/ed/files/documents/ai-report/ai-report.pdf

8. Wang S., et al. Artificial Intelligence in Education: A Systematic Literature Review // Expert Systems with Applications. – 2024. URL: https://www.sciencedirect.com/science/article/pii/S0957417424010339

9. Zhao Y., Pinto Llorente A. M., Cabero-Almenara J. Digital Competence in Higher Education Research: A Review // Computers & Education. – 2021. – Vol. 168. – P. 104211. – DOI: 10.1016/j.compedu.2021.104211.

УДК 004.8

Я. М. Мироненко *ст. 2 курсу*, Л. В. Константинова
jarik2707@gmail.com, liliyashel1976@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький, Україна

ОГЛЯД ТА ДОСЛІДЖЕННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

У сучасному світі, коли штучний інтелект поступово впроваджується в наше повсякденне життя, також зростає і залежність від його використання. Однак, через легкий доступ до цих технологій, з'являється ризик отримання неправдивої інформації зі штучно згенерованого контенту. Тому огляд та дослідження великих мовних моделей (LLM) може бути актуальним питанням в цей час для боротьби з подібними проблемами.

В роботі розглядаються питання ризиків та «сліпої довіри» до результатів роботи нейромереж, що є критичним аспектом, особливо в інженерії програмного забезпечення. В роботі відбувається пошук і виявлення слабких місць сучасних моделей, надаючи точні дані, щоб зрозуміти, які завдання можна безпечно передавати нейромережам, а які повинні обов'язково перевірятися фахівцем.

Метою дослідження є вивчення основ роботи великих мовних моделей та аналіз їх можливостей і обмежень. За сучасною класифікацію архітектур [1] великі мовні моделі використовують архітектуру трансформерів та глибоке машинне навчання (DL), що дозволяє ефективно аналізувати великі текстові масиви, а також будувати узагальнені представлення на основі проаналізованих даних.

Згідно з визначенням, наведеним у статті «Discover Applied Sciences» [1], великі мовні моделі – це клас нейромережевих алгоритмів які, базуються на архітектурі трансформерів, і використовують механізми самоуваги для виявлення складних зв'язків у тексті. Вони працюють як потужні інструменти прогнозування, що дозволяє їм створювати логічні та змістовні відповіді на запити користувачів. Сучасні моделі мають доступ до мільярдів параметрів, що дозволяє їм не лише запам'ятовувати інформацію, а й адаптовуватися до нових завдань без додаткового тренування. Це робить їх універсальними інструментами, які не обмежені певним набором попередньо встановлених команд.

У своєму аналізі Д. І. Мар'яш [2] зазначає, що на практиці ці системи інтегруються в робочі процеси, щоб автоматично генерувати контент, перекладати та скорочувати великі обсяги інформації. Важливим напрямом є сфера розробки програмного забезпечення, де моделі використовуються для перевірки, аналізу та оптимізації коду. Крім того, технологія дозволяє автоматизувати службу підтримки та виконати складний семантичний пошук у неструктурованих базах даних.

Навіть при великих можливостях, ці системи мають важливі обмеження, про що попереджають автори статті «On the Dangers of Stochastic Parrots» [3]. Основна проблема – це здатність до «галюцинацій», коли модель генерує вигідність, але неправильну інформацію. Їхні знання обмежені датою завершення навчання, а точність відповідей залежить від якості даних, на яких вони тренувалися. Також великою проблемою залишаються високі витрати на обчислювальні ресурси та ризики, пов'язані з безпекою і конфіденційністю.

Для практичної перевірки теоретичних висновків було проведено порівняльний аналіз роботи кількох провідних моделей від найвідоміших компаній. А саме, було розглянуто GPT-5.2 від OpenAI, Claude Sonnet 4.5 від Anthropic, Gemini 3 Pro від Google, Le Chat від Mistral AI (Mistral Large 2, Pixtral Large, Codestral, Les Ministraux, Mistral Saba), Grok 4.1 Thinking від xAI. Результати перевірки було продемонстровано у таблиці 1.

Для проведення тестування було розроблено власні складні тест-кейси, за допомогою яких здійснювалася перевірка кожної LLM. Створення тест-кейсів відбувалось за трьома ключовими критеріями: точність написання програмного коду, здатність до логічного мислення та стійкість до виникнення «галюцинацій» при роботі з неправдивими вхідними даними. Перевірка відбувалась наступними кроками:

1. Для тестування правильності написання програмного забезпечення було задано реалізувати інтерпретатор простої командної мови (DSL), який виконує команди над змінними та підтримує транзакції.

2. Для тестування логічних здібностей було задано вирішити наступну логічну задачу: «Визначити де знаходиться яблуко після того, як склянку з ним перевернули на стіл, а потім перемістили в інше місце».

3. Для перевірки схильності до «галюцинацій» було введено запитання, як використовувати (неіснуючу) функцію Adaptive AI Scanning в nmap.

Відповіді, що було отримано в результаті роботи було проаналізовано відповідно за моделями.

Не зважаючи на стрімкий розвиток LLM, навіть передові системи не застраховані від логічних помилок та генерації «галюцинацій» і тому їх застосування потребує перевірки.

Таблиця 1

Порівняльний аналіз роботи провідних великих мовних моделей

Критерії	GPT-5.2	Claude Sonnet 4.5	Gemini 3 Pro	Mistral	Grok 4.1
Якість коду	Коректна архітектура, містить критичні проблеми в обробці помилок	Повноцінний інкрементальний парсер, лише дрібні нюанси з валідацією	Помилки архітектури, потокового парсингу та в структурі вкладених commit/rollback	Неповна реалізація парсера, порушена інкапсуляція та некоректна структура вкладених транзакцій	Ефективна реалізація, відповідає вимогам задачі, з мінімальними компромісами в архітектурній масштабованості
Логічне мислення	Помилка у вирішенні, що яблуко перенесли в інше місце разом зі склянкою	Правильна відповідь, що яблуко випало з перевернутої склянки та залишилось на столі	Коректно вирішено, що яблуко залишилось на столі	Логічний ланцюжок слів привів нейромережу до хибної думки	Правильно вирішено, що яблуко залишилось на столі
Стійкість до «галюцинацій»	Відповідь була, що це не стандартна бібліотека але було вигадано інформацію до неіснуючої функції	Відразу отримано правильну відповідь, що такої функції не існує, а також наведено схожі за назвою альтернативи, які виявились правдивими	На початку заперечення, що функція є вбудованою, але потім почав будувати здогадки що це вбудований адаптивний механізм	Явно виражені галюцинації, вигадано, що функція інтегрована разом з Nmap	Відповідь, що функції не існує в офіційному інструменті Nmap, а також проаналізувавши джерела надано інформацію щодо існуючих можливостей

Аналіз результатів за моделями.

GPT-5.2 - модель продемонструвала непогане розуміння архітектури коду, проте припустилася критичних помилок у синтаксисі команд. У логічному завданні вона не змогла врахувати фізичний контекст, а при перевірці на «галюцинації» схильна додумувати інформацію, навіть ідентифікувавши відсутність стандартної бібліотеки.

Claude Sonnet 4.5 - ця нейромережа стала лідером тестування, показавши найкращий результат у написанні коду із глибоким розумінням технічних нюансів. Вона безпомилково впоралася із логічною загадкою та продемонструвала високу стійкість до «галюцинацій», запропонувавши реальні альтернативи замість вигаданих функцій.

Модель Gemini 3 Pro показала змішані результати: вона успішно розв'язала логічну задачу, але мала суттєві проблеми з семантикою коду та транзакціями. У тесті на «галюцинації» система коливалася, спочатку давши правильну відповідь, але згодом почала будувати хибні здогадки, що свідчить про нестабільність при роботі з невідомими даними.

Mistral виявилася найслабшою ланкою в цьому порівнянні. Модель не впоралася з жодним завданням: код був неробочим через порушення інкапсуляції, логічний ланцюжок привів до хибного висновку, а рівень «галюцинацій» виявився критичним - система вигадала неіснуючий функціонал.

Grok 4.1 продемонстрував високу стабільність і збалансованість. Модель надала ефективний код, правильно вирішила логічну задачу та показала відмінну роботу з джерелами, чітко вказавши на відсутність запитуваної функції без спроб ввести користувача в оману.

Висновки. Підсумовуючи проведене дослідження, варто зазначити, що великі мовні моделі є потужним інструментом сучасних інформаційних технологій, здатним значно пришвидшити розробку та аналіз даних. Проте результати тестування підтверджують, що навіть передові системи не застраховані від логічних помилок та генерації неправдивої інформації («галюцинацій»). Тому їх практичне застосування вимагає критичного підходу, обов'язкової верифікації результатів фахівцем та відповідального ставлення до обмежень кожної конкретної архітектури.

Список літератури

1. Large language models: an overview of foundational architectures, recent trends, and a new taxonomy. Discover Applied Sciences, Volume 7, article number 1027, 2025.

2. Д. І. Мар'яш Аналіз тексту з використанням великих мовних моделей. Київ: КПІ ім. І. Сікорського, 2024.

3. Bender E. M., Gebru T. et al. On the Dangers of Stochastic Parrots. FAccT, 2021.

УДК 004.8+614.8+331.452

К.М. Марченко, О.С. Гончарук, Д.М. Кучер, С.Г. Михайленко
k_marchenko@i.ua
Центральноукраїнський національний технічний університет, Кропивницький

ВПЛИВ ВПРОВАДЖЕННЯ ШТУЧНОГО ІНТЕЛЕКТУ НА БЕЗПЕЧНІСТЬ ПРАЦІ НА ПІДПРИЄМСТВІ

Штучний інтелект (ШІ) активно впроваджується у різних сферах виробничої діяльності. Разом з тим, виникає необхідність досліджень впливу впровадження ШІ на безпеку праці. Такі дослідження проводяться як вітчизняними тік і зарубіжними вченими та фахівцями з безпеки праці [1-5]. Певні ризики полягають у тому, що як і ці дослідження, так і розробка законодавчої та нормативної бази суттєво відстають від темпів впровадження ШІ. У багатьох випадках це впровадження відбувається стихійно й недостатньо контролюється з боку держави та органів, на які покладені повноваження з охорони праці [1].

Аналіз різних галузей та сфер виробництва показав, що впровадження ШІ в багатьох випадках не тільки покращують безпечність праці, а й кардинально вирішують певні проблеми. В системі безпеки праці ініціюється перехід від реакції на інцидент до його запобігання на основі даних.

Так, наприклад, безперервний моніторинг виробничого обладнання за допомогою аналізу датчиків ШІ дозволяє уникнути як поламок, аварій та травм, так і зниження якості продукції. За допомогою відеонагляду та аналізу образів ШІ виявляє відсутність засобів індивідуального захисту у працівників, в місцях, де їх наявність суворо регламентована. Камери з ШІ створюють віртуальні бар'єри навколо трансформаторів і відкритих розподільних пристроїв. При вході людини до небезпечної зони система миттєво подає сигнал або відключає живлення. ШІ наразі вміє розпізнавати пози робітників, виявляти їх втому, стрес, ментальний та психічний стан і, таким чином, попереджати травмування. Завдяки кейсу Field1st (2025) ШІ виявив, що втомлені бригади при температурі вище +30 °C в 3 рази частіше припускаються помилок, які ведуть до травм. Смарт-браслети виміряють пульс і температуру працівників, запобігаючи їх перегріву або перевтомі.

З іншого боку, впровадження ШІ породжує нові ризики та нові види небезпек. Серед них найкритичнішим ризиком є помилки в алгоритмах та функціонуванні ШІ, які можуть привести до хибних рішень, збоїв у функціонування систем, аварій та травм. Наприклад, збої в алгоритмах промислових роботів чи безпілотного транспорту можуть призвести до зіткнень із персоналом. Зміна темпу роботи під диктування алгоритмів підвищує ризик фізичної перевтоми. Існує тенденція деградації навичок співробітників при надмірній довірі до автоматики.

Висновки. Виконаний аналіз переваг та ризиків, пов'язаних з впровадженням штучного інтелекту у виробництво, показує необхідність

1. Виконання специфічних досліджень впливу впровадженням ШІ на безпеку працівників у кожному випадку з урахуванням особливостей виробничих процесів та умов праці.
2. Залишати критичні процеси під контролем експертів-людей для перевірки висновків системи.
3. Створення загальних принципів та правил безпеки праці з урахуванням дій ШІ, що може викликати формування окремої науки як галузі охорони праці.
4. Розробка вичерпної законодавчої та нормативної бази, що враховує всі аспекти впливу ШІ на працівників.

Список літератури

1. К.М. Марченко, О.В. Оришака, А.К. Марченко, А.М. Мельник. Ризики впровадження штучного інтелекту в комп'ютерні системи / Центральноукраїнський науковий вісник: Технічні науки, вип. № 4 (32), ч. 1. – Кропивницький, ЦНТУ, 2022 - с. 119-124.
2. Дрозд В. Можливості використання штучного інтелекту в охороні праці. 26.02.2026р. Джерело: https://pro-op.com.ua/article/17264-mozhlivosti-vikoristannya-shtuchnogo-intelektu-v-okhoroni-pratsi#ancex1.
3. Заплатинський В.М. Застосування штучного інтелекту для оцінки ризиків в охороні праці. Актуальні проблеми та перспективи розвитку охорони праці, безпеки життєдіяльності та цивільного захисту : мат-ли VII Міжнар. наук.- практ. конф. — Одеса : ОДАБА, 2025. – С. 85-87.
4. Joana Eva Dodoo 1, Hosam Al-Samarraie. Digital Innovations for Occupational Safety: Empowering Workers in Hazardous Environments. PMCID: PMC10928957 PMID: 38193448
5. Cynthia Chiles, Alan Gallacher. Global Workplace Safety Trends for 2025: How AI and Wearable Technology are Transforming Safety. URL: https://www.cc-global.com/blog/2025/global-workplace-safety-trends-for-2025-how-ai-and-wearable-technology-are-transforming-safety.

УДК 004.031 42

С.І. Зобенко[1], Р.М. Минайленко[2], А.С. Коваленко[2]
aron70@ukr.net
[1]Ліцей «Лідер», Кропивницький
[2]Центральноукраїнський національний технічний університет, Кропивницький

АНАЛІЗ РИЗИКІВ ВПРОВАДЖЕННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ОСВІТНІЙ ПРОЦЕС

На даний момент епоха антропогену в сучасній освіті поступається місцем алгоритмічному детермінізму. Стрімке проникнення генеративних моделей (Generative AI) у навчальний процес створює ситуацію, яку дослідники називають «педагогічним шоком». Головна небезпека полягає не в самій наявності технології, а в швидкості її адаптації, яка випереджає розвиток критичного мислення та етичних понять у здобувачів освіти. А академічна доброчесність трансформується з формального дотримання правил цитування у складну (з точки зору етики і не тільки її) стратегію виживання людського інтелекту.

Основним ризиком є так зване «явище когнітивного аутсорсингу». Коли здобувач для структурування творчої роботи або вирішення математичних задач використовує LLM (Large Language Models) відбувається процес заміщення нейронних зв'язків алгоритмічними послідовностями.

У текстах, згенерованих штучним інтелектом (ШІ), попри їхню граматичну досконаліть, відсутній суб'єктивний досвід та індивідуальний стиль. Наслідком постійної експлуатація таких інструментів буде «стилістична гомогенізація» — ситуація, коли роботи більшості здобувачів будуть ідентичними за логікою та лексичним запасом. Формуватиметься небезпечний психологічний ефект: здобувач буде вірити, що володіє знаннями, хоча насправді він лише володіє навичкою формулювання запиту. Це створить прошарок «фахівців» здатних лише до ретрансляції чужих ідей.

ШІ не копіює слова, він копіює концепції та структуру. Це породжує нову форму інтелектуального шахрайства, яке майже неможливо довести інструментально без порушення презумпції невинуватості здобувача. Поширення сервісів перевірки на ШІ-контент призводить до «полювання на відьом». Високий рівень помилкових спрацювань (false positives) руйнує етичний контракт між викладачем та здобувачем освіти і демотивує тих, хто пише самостійно, використовуючи сучасні цифрові інструменти для редагування.

Оскільки ШІ навчається на історичних даних, він часто відтворює застарілі гендерні, расові або культурні упередження. Безконтрольне використання таких моделей в гуманітарних дисциплінах загрожує регресом суспільної думки і кризою класичних форм контролю.

Педагоги змушені переходити від оцінки результату (тексту) до оцінки процесу (мислення). Це вимагає значного збільшення часу на усні співбесіди, дискусії та практичні проєкти, що в умовах масової освіти є надзвичайно трудомістким завданням.

Також ризиком є поява нового розриву між тими, хто має доступ до платних, досконаліших моделей ШІ і тими, хто користується безкоштовними застарілими версіями. Це порушує базовий принцип рівності можливостей в освіті.

Штучний інтелект в освіті — це не лише інструмент, а й віддзеркалення оцінок самої цінності зусиль, прикладених до пізнання. Для подолання цих викликів необхідна:

1. Етична сертифікація: Впровадження обов'язкового маркування використання ШІ в усіх наукових та навчальних роботах.
2. Реформа оцінювання: Пріоритет на метакогнітивні навички — здатність аналізувати, синтезувати та верифікувати інформацію, а не просто її відтворювати.
3. Критична ШІ-грамотність: Навчати здобувачів не навичкам обходу детекторів, а тому, як використовувати сучасні алгоритми для вдосконалення власних інтелектуальних здібностей без втрати суб'єктності.

Список літератури

1. Chaudhru, M. A., & Kazim, E. (2022). Artificial Intelligence in Education (AIEd): A high-level academic and industry note. AI and Ethics, (2), 157-165. doi: 10.1007/s43681-021-00074-z
2. Chen, L., Chen, P., & Lin, Z. (2020). Artificial Intelligence in Education: A Review. IEEE Access, (8), 75264-75278. doi: 10.1109/ACCESS.2020.2988510
3. Eynon, R (2015). The quantified learner: Discourses of data in education. Learning, Media and Technology, 40(1), 64-82. URL: http://www.tandfonline.com/doi/abs/10.1080/17439884 .2015.1100797
4. Farrokhnia, M., Banihashem, S. K., Noroozi, O., & Wals, A. (2023). A SWOT analysis of ChatGPT: Implications for educational practice and research, Innovations in Education and Teaching International. DOI: https://doi.org/10.1080/1470329 7.2023.2195846

УДК 004.312

Н.М.Якименко[1], Л.М.Поліщук[1]
yakymenkonm@kntu.kr.ua, pli_80@ukr.net
[1]*Центральноукраїнський національний технічний університет, Кропивницький*

ТЕХНОЛОГІЇ ЦИФРОВОЇ ЛОГІКИ ТА МОДЕЛЮВАННЯ: ВІД БАЗОВИХ ВЕНТИЛІВ ДО СУЧАСНИХ ОБЧИСЛЮВАЛЬНИХ АРХІТЕКТУР

Анотація У статті розглядається еволюція технологій цифрової логіки від фундаментальних логічних вентилів до високорівневих парадигм проєктування сучасних обчислювальних архітектур. Аналізується перехід від класичного RTL-моделювання до використання новітніх мов опису апаратури (наприклад, Chisel) та інтеграції штучного інтелекту в інструменти автоматизації електронного проєктування (EDA). Особлива увага приділяється предметно-орієнтованим архітектурам (DSA), відкритим стандартам на кшталт RISC-V та викликам апаратної безпеки в умовах субнанометрових технологічних норм.

1. Вступ

Сучасна цифрова епоха, що характеризується стрімким розвитком штучного інтелекту (ШІ), хмарних обчислень та Інтернету речей (IoT), висуває безпрецедентні вимоги до продуктивності, енергоефективності та безпеки обчислювальних систем. Незважаючи на те, що базовим будівельним блоком будь-якої цифрової системи залишається логічний вентиль (AND, OR, NOT), реалізований на базі транзисторів, методологія їх об'єднання у складні обчислювальні архітектури зазнала фундаментальних змін.

Сьогодні проєктування мікропроцесорів із мільярдами транзисторів неможливе без багаторівневих систем абстракції та складного математичного моделювання. Експоненційне зростання складності інтегральних схем вимагає нових підходів до дизайну цифрової логіки, що виходять за межі традиційного закону Мура.

2. Фізичні межі та еволюція базових елементів

Класична комп'ютерна інженерія базувалася на CMOS-технології (комплементарна структура метал-оксид-напівпровідник). Проте зі зменшенням технологічних норм до 5 нм, 3 нм і нижче, індустрія зіткнулася з квантовими ефектами, струмами витоку та проблемами розсіювання тепла.

Це призвело до революції на фізичному рівні цифрової логіки. На зміну площинним транзисторам прийшли FinFET (польові транзистори з каналом у вигляді плавця), а з 2022-2023 років провідні виробники (Samsung, TSMC, Intel) почали масовий перехід на архітектуру GAAFET (Gate-All-Around FET) та технологію подачі живлення зі зворотного боку кристала (Backside Power Delivery).

Зміна фізичної структури вентилів безпосередньо впливає на процеси моделювання. Сучасні інструменти TCAD (Technology Computer-Aided Design) змушені враховувати атомарну структуру матеріалів, що робить процес валідації цифрової логіки на найнижчому рівні надзвичайно ресурсомістким завданням.

3. Зміна парадигми апаратного моделювання: від RTL до ESL

Історично проєктування цифрових систем спиралося на рівень передачі регістрів (RTL — Register-Transfer Level) з використанням мов опису апаратури, таких як VHDL та SystemVerilog. Хоча ці стандарти залишаються індустріальною базою, вони стають занадто багатослівними та неефективними для опису гетерогенних багатоядерних систем.

Сьогодні спостерігається чіткий зсув у бік електронного системного рівня (ESL — Electronic System Level) та використання сучасних генераторів апаратного коду:

• Мови на базі Scala та Python: Інструменти нового покоління, такі як Chisel (Constructing Hardware in a Scala Embedded Language) та Amaranth/Migen (на базі Python), дозволяють застосовувати парадигми об'єктно-орієнтованого та функціонального програмування для генерації апаратного забезпечення. Це дозволяє параметризувати цифрову логіку на етапі компіляції та автоматично генерувати синтезований Verilog-код.

• Високорівневий синтез (HLS — High-Level Synthesis): Сучасні САПР (наприклад, Vivado HLS) дозволяють інженерам писати алгоритми на C/C++, які транслятор автоматично перетворює на оптимізовану цифрову логіку. Це критично важливо для швидкого прототипування алгоритмів цифрової обробки сигналів та комп'ютерного зору на базі ПЛІС (FPGA).

• Штучний інтелект у проєктуванні (AI in EDA): Починаючи з 2021 року, розробники систем автоматизації електронного проєктування впроваджують алгоритми навчання з підкріпленням (Reinforcement Learning) для розв'язання задачі трасування та розміщення елементів на кристалі (Floorplanning). Це дозволяє скоротити час моделювання топології з кількох місяців до кількох днів.

4. Архітектурні інновації: епоха спеціалізації

Уповільнення класичного масштабування продуктивності процесорів загального призначення (CPU) зумовило перехід до предметно-орієнтованих архітектур (Domain-Specific Architectures — DSA). Моделювання сьогодні фокусується не стільки на підвищенні тактової частоти універсальних конвеєрів, скільки на створенні спеціалізованої цифрової логіки під конкретні класи задач:

1. Тензорні та нейропроцесори (TPU/NPU): Архітектури, спеціально змодельовані для виконання матричних множень, що лежать в основі глибинного навчання. Їхня цифрова логіка оптимізована для систолічних масивів та роботи з даними зниженої точності (FP16, INT8, INT4), що значно підвищує енергоефективність.
2. Відкрита архітектура RISC-V: Відкритий стандарт системи команд (ISA) RISC-V став справжнім каталізатором інновацій в академічному та індустріальному середовищах. Завдяки відкритій ліцензії, дослідники можуть вільно моделювати, модифікувати та тестувати власні мікроархітектурні рішення, додаючи кастомні інструкції на рівні цифрової логіки без порушення патентних прав.
3. Адаптивні обчислювальні платформи (ACAP): Еволюція класичних ПЛІС (FPGA) призвела до появи гетерогенних кристалів, що об'єднують класичну програмовану цифрову логіку, хардверні ARM-ядра та спеціалізовані ШІ-рушії в єдиній системі на кристалі (SoC), що вимагає комплексних підходів до апаратно-програмного ко-моделювання.

5. Апаратна безпека як невід'ємна частина моделювання

В умовах глобальних кіберзагроз інформаційна безпека змістилася з програмного рівня на апаратний. Сучасні методології проєктування цифрової логіки обов'язково включають моделювання безпеки (Security-Aware Design):

- Апаратні трояни (Hardware Trojans): Зловмисні зміни, внесені в логіку кристала на етапі виробництва. Виявлення таких закладок вимагає складного формального моделювання та аналізу побічних каналів (енергоспоживання, електромагнітного випромінювання).
- Захищені анклави (Root of Trust): Моделювання ізольованих криптографічних співпроцесорів на рівні кремнію, які відповідають за безпечне завантаження, управління ключами та захист від фізичного втручання.
- PUF-технології (Physically Unclonable Functions): Використання мікроскопічних варіацій при виробництві логічних вентилів для створення унікального, неклонованого "відбитка пальця" інтегральної схеми.

Висновки

Технології цифрової логіки та апаратного моделювання переживають період наймасштабніших трансформацій за останні десятиліття. Перехід від створення простих комбінаційних схем до проєктування мільярдних гетерогенних систем вимагає безпрецедентного рівня абстракції.

Інтеграція мов програмування високого рівня (Scala, Python) у процеси апаратного опису, впровадження методів штучного інтелекту в маршрути проєктування (EDA) та стрімкий розвиток відкритих архітектур на базі RISC-V формують нове обличчя комп'ютерної інженерії. Водночас виклики фізичних обмежень на субнанометровому рівні та загрози апаратній безпеці стимулюють наукову спільноту до розробки принципово нових методів верифікації та моделювання обчислювальних систем майбутнього.

Список літератури

1. Mirhoseini, A., Goldie, A., Yazdanbakhsh, A., et al. (2021). A graph placement methodology for fast chip design. Nature, 594, 207-212.
2. LaMeres, B. J. (2019). Introduction to Logic Circuits & Logic Design with Verilog (2nd ed.). Springer.
3. Waterman, A., & Asanović, K. (Eds.). (2021). The RISC-V Instruction Set Manual, Volume I: Unprivileged ISA. RISC-V International.
4. Bhunia, S., & Tehranipoor, M. (2022). Hardware Security: A Hands-on Learning Approach (2nd ed.). Morgan Kaufmann.
5. Bailey, D. (2023). SystemVerilog for Hardware Description: RTL Design and Verification. Oxford University Press..

УДК 004.8:658.5

К.О. Ярмоленко, *студентка 4 курсу*
kateyarmolenko15@gmail.com
Науковий керівник: С.В. Науменко, *викладач*
Національний університет імені Богдана Хмельницького, Черкаси

ЗАСТОСУВАННЯ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ФОРМУВАННЯ КВІТКОВИХ БУКЕТІВ У БІЗНЕСІ

У сучасних умовах цифрової трансформації бізнесу актуальним є впровадження інтелектуальних систем, здатних автоматизувати процеси взаємодії з клієнтами та підтримки прийняття рішень. Одним із перспективних напрямів є використання генеративного штучного інтелекту для персоналізації продуктів і послуг. Особливо це актуально для малого та середнього бізнесу у сфері флористики, де значна частина рішень залежить від суб'єктивного досвіду фахівця.

У роботі розглядається підхід до використання генеративного штучного інтелекту в системі підбору та генерації квіткових букетів, інтегрованій у Telegram-бот. Запропонована система дозволяє формувати варіанти букетів на основі текстових запитів користувача із застосуванням методів обробки природної мови (NLP) та великих мовних моделей (LLM), що забезпечують інтерпретацію намірів користувача та генерацію релевантних композицій.

Архітектура запропонованої системи включає клієнтський рівень (Telegram-бот), серверну частину (API для обробки запитів) та модуль генеративного штучного інтелекту. Взаємодія між компонентами забезпечує обробку запитів у реальному часі та генерацію персоналізованих варіантів букетів без необхідності залучення консультанта. На відміну від традиційних підходів, де підбір букетів здійснюється вручну, запропонована система автоматизує процес формування композицій та знижує залежність від людського фактора. Це дозволяє скоротити час обробки запитів та підвищити якість обслуговування клієнтів.

Ефективність впровадження генеративного штучного інтелекту оцінюється за інтегральним показником:

$$E = \alpha C + \beta T + \gamma U + \delta Q,$$

де C — зниження операційних витрат, T — скорочення часу обробки замовлення, U — рівень задоволеності користувачів, Q — якість сформованих рішень; $\alpha,\beta,\gamma,\delta$ — вагові коефіцієнти.

Таблиця 1

Порівняння традиційного та автоматизованого підходів

Показник	Традиційний підхід	Запропонована система
Час обробки замовлення	10–15 хв	3–5 хв
Рівень персоналізації	Середній	Високий
Залежність від персоналу	Висока	Низька
Доступність сервісу	Обмежена	24/7

Разом з тим, впровадження генеративного штучного інтелекту має певні обмеження. До них належать залежність результатів від якості вхідних даних, можливість генерації некоректних варіантів та необхідність додаткової перевірки результатів. Це вимагає впровадження механізмів валідації та контролю якості.

Таким чином, використання генеративного штучного інтелекту у сфері флористики є ефективним інструментом автоматизації бізнес-процесів та підвищення якості обслуговування клієнтів. Запропонований підхід може бути адаптований до інших сфер, що потребують персоналізації продуктів і послуг.

Список літератури

1. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. Pearson, 2021.
2. Davenport T., Ronanki R. Artificial Intelligence for the Real World. Harvard Business Review, 2018.
3. Brynjolfsson E., McAfee A. The Second Machine Age. W.W. Norton & Company, 2014.

УДК 004.75:004.8

В.А. Лісовий[1], А.С. Коваленко[1]
lisovyi.vyacheslav.andriyovych@gmail.com, annasun911@gmail.com
[1]Центральноукраїнський національний технічний університет, Кропивницький

ОГЛЯД СУЧАСНИХ ПІДХОДІВ ДО АДАПТИВНОГО БАЛАНСУВАННЯ НАВАНТАЖЕННЯ З УРАХУВАННЯМ РЕСУРСНОЇ ОБУМОВЛЕНОСТІ

Стрімке зростання інтенсивності обчислень у хмарних, мультихмарних і периферійних середовищах, поширення контейнеризації, мікросервісної архітектури та сервісно орієнтованих платформ посилює значущість задачі ефективного балансування навантаження. У цих умовах продуктивність інформаційної системи визначається не лише рівномірністю розподілу запитів між вузлами, а й здатністю механізму керування враховувати поточний ресурсний стан інфраструктури, часову мінливість навантаження, якість обслуговування, обмеження SLA, енергоспоживання та вартість використання ресурсів. Саме тому в сучасних дослідженнях балансування навантаження розглядається не ізольовано, а в тісному зв'язку із задачами планування виконання, розміщення та перерозподілу обчислювальних ресурсів у розподіленому середовищі.

Результати опрацювання наукових джерел дають підстави стверджувати, що в задачах адаптивного балансування навантаження та суміжних задачах планування ресурсів у хмарних середовищах і надалі застосовуються класичні, евристичні, метаевристичні, ML-орієнтовані та гібридні підходи. Зокрема, огляд [1] систематизує еволюцію технологій балансування навантаження та планування задач у контексті хмарних обчислень, а в огляді [2] показано зростання ролі методів машинного і глибинного навчання в цих задачах.

Важливим напрямом еволюції сучасних підходів є врахування ресурсної обумовленості під час прийняття рішень щодо перерозподілу навантаження. У праці [3] запропоновано багаторесурсно-орієнтовану стратегію (multi resource aware strategy) для гетерогенних периферійних (Edge) кластерів, у якій динамічно відстежуються залишкові потужності CPU, пам'яті, дискової та мережевої підсистем введення-виведення і зіставляються з ресурсними профілями вхідних задач. Подібна логіка простежується і в дослідженнях із суміжної сфери адаптивного планування ресурсів, зокрема в [5], де адаптивне планування в мультихмарному середовищі поєднує LSTM-прогнозування навантаження з багатокритеріальною оптимізацією розміщення задач із урахуванням енергоспоживання, затримок, дотримання SLA та завантаження вузлів. Отже, аналіз джерел дає підстави стверджувати, що сучасні підходи до балансування навантаження все частіше поєднують багаторесурсний аналіз із прогнозуванням стану інфраструктури, у межах якого рішення формуються не лише на основі поточного завантаження, а й на основі очікуваної динаміки стану інфраструктури.

Наступний етап розвитку адаптивного балансування навантаження та пов'язаних із ним процедур планування пов'язується з їх інтелектуалізацією на основі навчання з підкріпленням і споріднених підходів. У роботі [7] запропоновано ієрархічну модель на основі глибинного навчання з підкріпленням (DRL framework) для планування задач у хмарному середовищі, у межах якої рішення приймаються на двох рівнях, спочатку на рівні кластера віртуальних машин, а потім на рівні конкретної VM, що дає змогу адаптувати політику планування до масштабних і змінних хмарних середовищ. У статті [8] представлено модель динамічного балансування навантаження, яка поєднує часові графові нейронні мережі для прогнозування майбутнього стану ресурсів, імпульсні нейронні мережі для адаптивного прийняття рішень і навчання з підкріпленням для коригування політики розподілу на основі зворотного зв'язку. У сукупності ці праці свідчать про перспективність поєднання прогнозування, представлення складного стану середовища та адаптивного навчання політики керування.

Особливий інтерес для подальшого розвитку галузі становить використання великих мовних моделей як засобу підтримки прийняття рішень у задачах планування і балансування навантаження. У статті [4] запропоновано підхід на основі великої мовної моделі (LLM based framework LarS), у якому велика мовна модель виконує функцію високорівневого агента прийняття рішень під час планування задач у хмарному середовищі, а DRL-агент використовується для перевірки й відбору якісних траєкторій рішень. У публікації [6] показано близький за логікою підхід, де LLM спрямовує роботу SARSA-алгоритму через формування евристичної оцінки Q-функції, що підвищує адаптивність планування та зменшує ризик потрапляння в субоптимальні рішення. Це дає підстави розглядати великі мовні моделі як перспективний засіб інтеграції різнорідної інформації в задачах адаптивного керування навантаженням, що відкриває перспективу побудови контекстно чутливих механізмів адаптивного балансування навантаження.

Отже, аналіз джерел [1-8] дає підстави стверджувати, що сучасні дослідження у сфері адаптивного балансування навантаження та пов'язаних із ним задач планування ресурсів розвиваються за трьома взаємопов'язаними напрямами, а саме переходом від класичних евристичних методів до інтелектуальних адаптивних алгоритмів, посиленням уваги до багаторесурсної обумовленості середовища та інтеграцією прогнозних і LLM-орієнтованих механізмів прийняття рішень. Водночас аналізовані праці переважно висвітлюють окремі аспекти цієї проблематики і не формують цілісного підходу, який би поєднував аналіз з урахуванням ресурсного стану середовища, прогнозування стану інфраструктури, адаптивне балансування навантаження та використання великих мовних моделей як інтелектуального компонента керування. На рис. 1

подано концептуальну схему сучасних підходів до адаптивного балансування навантаження з урахуванням ресурсної обумовленості та їх інтеграції в межах перспективного цілісного підходу. Це підсилює наукову й практичну актуальність дослідження, спрямованого на розроблення інформаційної технології адаптивного балансування навантаження з урахуванням ресурсної обумовленості на основі великих мовних моделей.

Рисунок 1 – Концептуальна схема підходів до адаптивного балансування навантаження з урахуванням ресурсної обумовленості

Список літератури

1. Devi N., Dalal S., Solanki K. et al. A systematic literature review for load balancing and task scheduling techniques in cloud computing. Artificial Intelligence Review. 2024. URL: https://doi.org/10.1007/s10462-024-10925-w (дата звернення: 20.04.2026).

2. Sonia, Nath R. A systematic review of various load balancing approaches in cloud computing utilizing machine learning and deep learning. International Journal of Data Science and Analytics. 2025. URL: https://doi.org/10.1007/s41060-025-00718-x (дата звернення: 20.04.2026).

3. Li X., Zhu E., Qin J. et al. A Multi-Resource-Aware and Load-Balanced Scheduling Strategy for Heterogeneous Edge Clusters. Journal of Grid Computing. 2025. URL: https://doi.org/10.1007/s10723-025-09817-2 (дата звернення: 20.04.2026).

4. Pei H., Gu Y., Sun Y. et al. LLM-based cost-aware task scheduling for cloud computing systems. Journal of Cloud Computing. 2025. URL: https://doi.org/10.1186/s13677-025-00822-0 (дата звернення: 20.04.2026).

5. Sefati S. S., Keymasi M., Craciunescu R. et al. Adaptive Resource Scheduling in Multi-Cloud Computing Using Recurrent Neural Forecasting and Memory-Based Metaheuristic Optimization. Journal of Grid Computing. 2025. URL: https://doi.org/10.1007/s10723-025-09812-7 (дата звернення: 20.04.2026).

6. Krishnamurthy B., Shiva S. G. Large Language Model-Guided SARSA Algorithm for Dynamic Task Scheduling in Cloud Computing. Mathematics. 2025. Vol. 13, no. 6. Art. 926. URL: https://doi.org/10.3390/math13060926 (дата звернення: 20.04.2026).

7. Cui D., Peng Z., Li K. et al. An novel cloud task scheduling framework using hierarchical deep reinforcement learning for cloud computing. PLOS ONE. 2025. Vol. 20, no. 8. Art. e0329669. URL: https://doi.org/10.1371/journal.pone.0329669 (дата звернення: 20.04.2026).

8. Rajammal K., Chinnadurai M. Dynamic load balancing in cloud computing using predictive graph networks and adaptive neural scheduling. Scientific Reports. 2025. URL: https://doi.org/10.1038/s41598-025-97494-2 (дата звернення: 20.04.2026).

УДК 004

С.В. Богаченко, В.Р. Меркулов,
serg123bb@gmail.com, mvladyslavr@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

ДОСЛІДЖЕННЯ ТА РОЗРОБКА МЕТОДІВ ПІДВИЩЕННЯ ЕНЕРГОЕФЕКТИВНОСТІ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ В УМОВАХ ОБМЕЖЕНИХ РЕСУРСІВ

Постановка задачі. Задачі машинного навчання стрімко набувають поширення в різних галузях сучасного суспільства. Активний розвиток цієї сфери зумовлений як збільшенням обсягів даних, так і появою нових алгоритмів та обчислювальних підходів. Особливої актуальності набуває впровадження моделей штучного інтелекту у системах з обмеженими ресурсами, таких як мобільні пристрої, вбудовані системи (embedded systems), IoT-пристрої, автономні сенсори та дрони. У таких умовах ключовим фактором стає не лише точність моделей, але й їх енергоефективність. Таким чином, проблема зниження енергоспоживання моделей машинного навчання без значної втрати точності є критично важливою.

Основна частина. Сучасні моделі глибокого навчання, зокрема згорткові та трансформерні архітектури, характеризуються високою обчислювальною складністю. Їх використання у ресурсно-обмежених середовищах призводить до швидкого розряду батареї, перегріву пристроїв та зниження продуктивності. У зв'язку з цим виникає необхідність розробки методів оптимізації, спрямованих на зменшення обчислювального навантаження. Основною метою даної роботи є дослідження та розробка методів підвищення енергоефективності моделей машинного навчання в умовах обмежених ресурсів.

Одним із ключових напрямків є використання алгоритмів оптимізації моделей. До таких методів належать pruning (обрізання ваг), quantization (квантування), knowledge distillation (дистиляція знань) та використання компактних архітектур (наприклад, MobileNet, EfficientNet). Обрізання дозволяє видалити надлишкові параметри моделі, що незначно впливають на результат, зменшуючи розмір мережі. Квантування знижує точність представлення ваг (наприклад, з float32 до int8), що значно зменшує обсяг пам'яті та енергоспоживання. Застосування цих підходів дозволяє досягти суттєвого зниження енергоспоживання без критичного погіршення якості моделі.

Важливу роль відіграє також вибір апаратної платформи. Сучасні процесори та прискорювачі, такі як TPU, NPU або енергоефективні GPU, оптимізовані для виконання операцій машинного навчання. Використання спеціалізованих інструкцій та апаратних блоків дозволяє значно знизити енергоспоживання порівняно з універсальними CPU. Оптимальне поєднання програмних та апаратних засобів є ключем до побудови енергоефективних систем.

Для експериментальної перевірки було розроблено програмний модуль, який реалізує оцінку енергоспоживання моделей машинного навчання на різних етапах оптимізації. Методика передбачає вимірювання часу виконання, споживаної потужності та обчислення показника energy per inference (енергія на один прогноз). В якості тестових моделей було використано як базові нейронні мережі, так і їх оптимізовані версії після застосування pruning та quantization. Такий підхід дозволяє отримати об'єктивну оцінку ефективності кожного методу оптимізації.

Результати експерименту показали, що базові моделі без оптимізації демонструють високий рівень енергоспоживання, що робить їх непридатними для використання на пристроях з обмеженим живленням. Після застосування квантування спостерігалося зниження енергоспоживання на 30–50%, тоді як обрізання моделі дозволило зменшити обчислювальні витрати ще на 20–30%. Комбіноване застосування методів дало найкращий результат. Загальний виграш в енергоефективності склав до 60% при збереженні прийнятного рівня точності.

Порівняльний аналіз показав, що використання оптимізованих моделей дозволяє значно продовжити час автономної роботи пристроїв, що є критичним для мобільних та вбудованих систем. Крім того, зменшення обчислювального навантаження знижує тепловиділення, що позитивно впливає на стабільність роботи обладнання. Таким чином, енергоефективне машинне навчання є ключовим фактором для розвитку сучасних інтелектуальних систем.

Висновки. У висновках зазначено, що впровадження методів оптимізації моделей є ефективним підходом до зниження енергоспоживання в умовах обмежених ресурсів. Розроблений підхід дозволяє адаптувати складні моделі машинного навчання для використання у реальних системах з обмеженим енергобюджетом. Перспективи подальших досліджень полягають у використанні адаптивних моделей, які динамічно змінюють свою складність залежно від доступних ресурсів, а також у дослідженні нових енергоефективних архітектур нейронних мереж.

УДК 004.738.5:796.323

Д.О. Красільніков *студент 4 курсу, група КН-22*
Спеціальність 122 «Комп'ютерні науки»
E-mail: krasilnikov.daniil1122@vu.cdu.edu.ua
Науковий керівник: Б.В. Мисник*, к.т.н.*
Черкаський національний університет імені Богдана Хмельницького, Черкаси

ІНФОРМАЦІЙНА ВЕБСИСТЕМА ДЛЯ НАВЧАННЯ БАСКЕТБОЛУ

Баскетбол є одним із найпопулярніших командних видів спорту у світі, яким захоплюються понад 450 мільйонів людей у більш ніж 200 країнах. Незважаючи на широку популярність цього виду спорту, доступ до якісного навчання залишається нерівномірним. Більшість початківців, особливо у невеликих містах і сільській місцевості, позбавлені можливості займатися під керівництвом досвідченого тренера через географічну віддаленість спортивних шкіл, значну вартість індивідуальних занять та часові обмеження, пов'язані з необхідністю фізичної присутності на тренуваннях.

Стрімкий розвиток вебтехнологій відкриває принципово нові можливості для організації дистанційної спортивної освіти. Передача знань, відпрацювання техніки та персональний супровід учня стають можливими без прив'язки до конкретного місця й часу. Проте наявні зарубіжні платформи онлайн-навчання баскетболу не задовольняють потреби української аудиторії повною мірою: одні пропонують лише відеоконтент без індивідуального супроводу тренера, інші зосереджені виключно на окремих аспектах гри або є недоступними через мовний бар'єр чи завищену ціну. Жодна з них не забезпечує комплексного підходу.

Головна наукова ідея роботи полягає в обґрунтуванні того, що ефективне дистанційне навчання баскетболу є досяжним лише тоді, коли в єдиній інформаційній системі органічно поєднані три взаємозалежні складові. Перша складова – це структурований мультимедійний навчальний контент, організований відповідно до рівня підготовки учня. Друга складова – це механізм персоналізації тренувального процесу, що враховує індивідуальні цілі, темп опанування матеріалу та поточний прогрес. Третя складова – це інструменти живої комунікації між студентом і тренером, які забезпечують своєчасний зворотний зв'язок і можливість коригування навчального маршруту.

Метою роботи є проєктування та розробка інформаційної вебсистеми для навчання баскетболу, яка усуває ключові недоліки наявних рішень і надає учням доступ до якісної спортивної освіти незалежно від місця проживання, рівня попередньої підготовки та фінансових можливостей. Система має забезпечити повноцінний освітній процес, порівнянний за якістю з очним навчанням у спортивній секції, проте позбавлений притаманних йому обмежень.

У результаті дослідження спроєктовано вебсистему, що реалізує повний цикл дистанційного навчання баскетболу. Учень самостійно обирає курс відповідно до власного рівня підготовки та переглядає структуровані відеоуроки у зручний для себе час, маючи змогу повернутися до будь-якого з них необмежену кількість разів. Тренер формує для кожного учня індивідуальний план занять із конкретними завданнями та термінами їх виконання, відстежує навчальний прогрес і своєчасно вносить корективи. Вбудована система обміну повідомленнями замінює живу комунікацію в умовах дистанційного навчання і дозволяє тренеру надавати детальний зворотний зв'язок.

Розроблена система вирішує проблему фрагментарності освітнього процесу, характерну для наявних платформ. Учень рухається за чітко визначеним індивідуальним навчальним маршрутом, бачить власний прогрес і відчуває постійну підтримку тренера. Це наближає дистанційне навчання до очного за ефективністю. Тренер, зі свого боку, отримує інструмент для одночасної роботи з широкою аудиторією учнів з будь-якої точки країни, що суттєво розширює можливості для його професійної діяльності.

Наукова новизна роботи полягає в обґрунтуванні та практичній реалізації підходу до організації дистанційного спортивного навчання, за якого персоналізація освітньої траєкторії та безпосередня взаємодія з тренером є обов'язковими системними елементами, а не другорядними додатковими функціями. Такий підхід дозволяє максимально наблизити дистанційне навчання до індивідуального очного тренування, зберігаючи при цьому всі переваги онлайн-формату.

Практичне значення отриманих результатів полягає у тому, що розроблена система може бути застосована як тренерами-індивідуалами для масштабування своєї педагогічної діяльності та охоплення ширшої аудиторії, так і спортивними школами та федераціями для організації системного дистанційного навчання.

Список літератури

1. FIBA. Basketball worldwide: official statistics. URL: https://www.fiba.basketball

2. Олексієнко О.І. Проектування веб-орієнтованих інформаційних систем: навч. посіб. Черкаси: ЧНУ, 2021. 210 с.

ЗМІСТ

СЕКЦІЯ 1. ІНФОРМАЦІЙНА БЕЗПЕКА ДЕРЖАВИ, СУСПІЛЬСТВА ТА ОСОБИСТОСТІ

СЕКЦІЯ 2. ПРОГРАМУВАННЯ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

СЕКЦІЯ 3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ, МЕДИЦИНІ ТА ОСВІТІ

НАУКОВЕ ВИДАННЯ

ТЕЗИ ДОПОВІДЕЙ

IX Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні технології"

23 квітня 2026 року

Матеріали публікуються в авторській редакції.
За достовірність викладених фактів, цитат та інших відомостей відповідальність несуть автори.

Відповідальний за випуск: *О.А. Смірнов*

Комп`ютерна верстка: *Р.М. Минайленко*

Електронне видання

Центральноукраїнський національний технічний університет
пр-кт Університетський, 8, м. Кропивницький, 25006.
тел. (0522) 559-245, www.kntu.kr.ua