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

УДК 004.4

Тези доповідей

VIII Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні технології"

24-25 квітня 2025 року

Кропивницький 2025

УДК 004.4

Матеріали VIII Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології": тези доповідей, 24-25 квітня 2025 р. – Кропивницький: ЦНТУ, 2025. – 97 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 004.056.5, 004.272

М.О. Ларченко
urlinka2006@gmail.com
Національний університет «Чернігівська політехніка», Чернігів
Ніжинський державний університет імені Миколи Гоголя, Ніжин

ЗАХИСТ ОПЕРАТИВНОЇ ПАМ'ЯТІ ЯК КЛЮЧОВИЙ ЕЛЕМЕНТ ДЕРЖАВНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

У сучасних умовах розвитку інформаційних технологій та зростання кількості кіберзагроз захист інформаційних систем держави є критично важливим для забезпечення національної безпеки. Одним із найбільш уразливих елементів інфраструктури є оперативна пам'ять (RAM), яка містить важливі дані, що використовуються в процесі обробки інформації. Атаки на оперативну пам'ять можуть призвести до витоку конфіденційних даних, що мають стратегічне значення для державних установ. Це дослідження висвітлює сучасні методи захисту оперативної пам'яті в контексті державної інформаційної безпеки, зокрема в умовах гібридних загроз.

Оперативна пам'ять є одним із найменш захищених компонентів комп'ютерних систем, оскільки в ній зберігаються тимчасові дані, які використовуються під час роботи з програмами. Існує кілька типів атак на пам'ять, зокрема:

1) RAM scraping – методика вилучення даних безпосередньо з оперативної пам'яті. Зловмисники використовують спеціалізовані інструменти для отримання конфіденційної інформації, яка тимчасово зберігається в пам'яті, наприклад, паролів чи інших автентифікаційних даних;

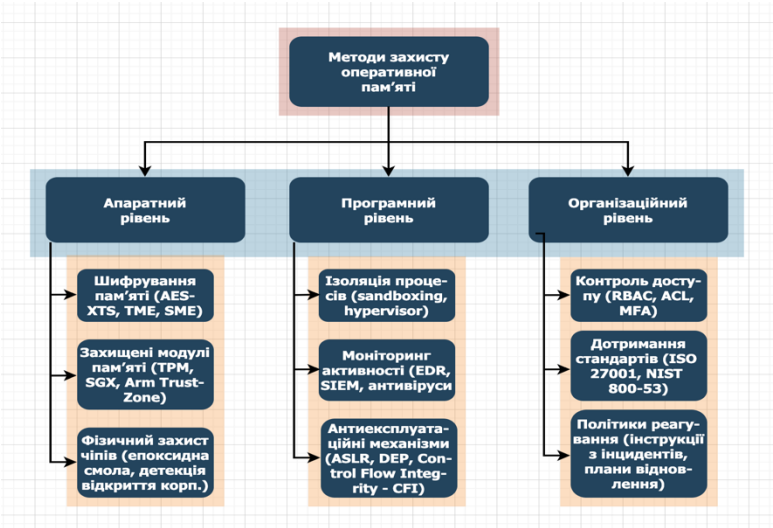
2) Cold boot attack – атака, при якій інформація з оперативної пам'яті витягується через фізичне вимкнення комп'ютера. Після цього зловмисники намагаються відновити дані за допомогою спеціальних методів, наприклад, при заморожуванні пам'яті або швидкому збереженні даних на зовнішньому носії;

3) Direct Memory Access (DMA) – атака, яка використовує порти введення/виведення (I/O порти) для безпосереднього доступу до пам'яті без залучення центрального процесора. Цей тип атак став можливим завдяки наявності сучасних компонентів, таких як FireWire або Thunderbolt, що дозволяють стороннім пристроям отримувати доступ до пам'яті комп'ютера [1].

Наслідки таких атак можуть бути катастрофічними: витік конфіденційних даних, порушення цілісності операційних систем, знищення або підміна критичних даних, які є основою державних інформаційних систем.

У зв'язку з високою уразливістю оперативної пам'яті до атак, важливою складовою заходів кібербезпеки є її захист. Для забезпечення конфіденційності та цілісності даних в оперативній пам'яті використовуються різні методи. Серед них апаратне шифрування пам'яті, що є одним із найбільш ефективних способів захисту даних. Це включає в себе використання спеціальних чипів для шифрування даних безпосередньо в процесі їх зберігання в пам'яті. Одним із таких стандартів є Intel's Total Memory Encryption (TME) та AMD Memory Encryption [2], які забезпечують автоматичне шифрування всіх даних у пам'яті на рівні апаратного забезпечення. Шифрування даних на рівні пам'яті дозволяє значно ускладнити доступ зловмисників до конфіденційної інформації, навіть якщо вони зможуть фізично дістатися до пам'яті (рис. 1).

Рис. 1. Блок-схема методів захисту оперативної пам'яті у вигляді дерева рішень



Іншим методом є верифікація цілісності даних у пам'яті. Вона включає використання методів для перевірки, чи були дані змінені або пошкоджені. Це можуть бути криптографічні підписи або контрольні суми, які дозволяють виявляти несанкціоновані зміни в даних. Одним з підходів є Memory Integrity функція в рамках Windows Defender. Вона гарантує, що критичні системні файли не були змінені в процесі їх обробки.

Ще один підхід – це віртуалізація та ізоляція пам'яті. Сучасні технології віртуалізації дозволяють ізолювати пам'ять для різних віртуальних машин, що є особливо корисним для забезпечення захисту державних інформаційних систем.

Ізоляція пам'яті між різними процесами та користувачами ускладнює атаку на пам'ять, оскільки кожен процес працює у своїй власній ізольованій області. Технології, такі як Intel SGX [3] та AMD SEV, дозволяють створювати захищене середовище для виконання кодів, що зменшує ризик витоку даних навіть у випадку зламу основної системи.

Стандарти, розроблені міжнародними організаціями, такими як NIST (Національний інститут стандартів і технологій США), мають велике значення для забезпечення безпеки пам'яті в державних інформаційних системах. Зокрема, можна згадати наступні чинні стандарти:

FIPS 140-3 – стандарт, що визначає вимоги до криптографічної апаратури та програмного забезпечення, які можуть бути використані для захисту даних, зокрема в оперативній пам'яті [4].

NIST SP 800-193 – стандарт, що охоплює захист інформаційних систем від атак на їхнє апаратне забезпечення, зокрема на пам'ять, за допомогою спеціалізованих механізмів для безпечного завантаження [5].

Ці стандарти допомагають державам створювати ефективні стратегії кібербезпеки, які відповідають сучасним вимогам і загрозам (табл. 1).

Таблиця 1

Порівняльна таблиця безпекових стандартів оперативної пам'яті

Стандарт	Основні вимоги	Галузь застосування	Використання в державних системах
FIPS 140-3(Federal Information Processing Standard)	Визначає вимоги до криптографічних модулів, включаючи шифрування пам'яті, фізичний захист та механізми контролю доступу.	Захист інформації в урядових системах, фінансових структурах, критичній інфраструктурі.	Обов'язковий для використання в державних установах США та рекомендований у міжнародній практиці.
NIST SP 800-193 (Platform Firmware Resiliency Guidelines)	Визначає механізми захисту прошивки від атак, виявлення компрометації та відновлення.	Безпечне завантаження операційних систем, виявлення несанкціонованих змін у мікропрограмному забезпеченні.	Використовується у військових і державних інформаційних системах для захисту від модифікацій на рівні платформи.

Для виявлення і попередження атак на оперативну пам'ять важливе значення мають інструменти аналізу пам'яті. Один з таких інструментів – Volatility Framework, який дозволяє проводити аналіз дампів пам'яті та виявляти ознаки компрометації системи.

Інші інструменти, такі як LiME (Linux Memory Extractor) та Rekall, використовуються для вилучення і аналізу даних з оперативної пам'яті в реальному часі. Вони дозволяють оперативно реагувати на потенційні загрози та виявляти наявність шкідливих процесів.

Таким чином, захист оперативної пам'яті є важливою складовою кібербезпеки державних інформаційних систем. Сучасні методи захисту, такі як апаратне шифрування, верифікація цілісності даних, віртуалізація пам'яті та використання стандартів безпеки, дозволяють значно зменшити ризики витоку або компрометації критичних даних. Однак для забезпечення надійного захисту необхідно комплексно підходити до проблеми, враховуючи як технічні, так і організаційні аспекти. Враховуючи швидкий розвиток кіберзагроз, важливо постійно вдосконалювати системи захисту оперативної пам'яті для забезпечення національної безпеки в умовах сучасних викликів.

Список літератури

1. Windows Kernel DMA Protection, Microsoft Security Blog. URL: https://learn.microsoft.com/en-us/windows/security/hardware-security/kernel-dma-protection-for-thunderbolt.

2. AMD Secure Memory Encryption (SME) – Technical Overview, Advanced Micro Devices. URL: https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/memory-encryption-white-paper.pdf

3. Intel Software Guard Extensions (SGX) – Developer Guide, Intel Corporation. URL: https://www.intel.com/content/www/us/en/content-details/671334/intel-software-guard-extensions-intel-sgx-developer-guide.html

4. Federal Information Processing Standard (FIPS) 140-3: Security Requirements for Cryptographic Modules. 2019. URL: https://csrc.nist.gov/pubs/fips/140-3/final

5. National Institute of Standards and Technology. NIST Special Publication 800-193. Platform Firmware Resiliency Guidelines. 2018. URL: https://csrc.nist.gov/pubs/sp/800/193/final

UDC 004.738.5:316.4

Oleksandr Dorenskyi, Viktor Kolesnyk
dorenskyiop@kntu.kr.ua, kolesnykviktor2006@gmail.com
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine

AI-BASED FACT-CHECKING METHOD FOR COUNTERING DISINFORMATION IN CYBERSPACE

In today's information space, the issue of disinformation is extremely significant. Due to information overload, it is often difficult for individuals to identify disinformation on their own. This issue has become especially critical amid Russian aggression. Under these circumstances, artificial intelligence technologies serve as powerful tools both for generating and detecting disinformation within the digital information environment.

According to the International Economic Forum (2024), disinformation has ranked first among short-term global risks. Generative AI is responsible for up to 30% of fake content on social media platforms, while 50% of users cannot distinguish deepfakes from authentic materials. Key trends include increased automation, personalization of attacks, and hybrid campaigns [1-3].

The research presented in [4] and [5] addresses the pressing issue of improving the effectiveness of counter-disinformation efforts within information warfare, which in Ukraine is carried out by the Center for Countering Disinformation. However, in the coordination model of the information warfare system presented in [4], little attention is given to fact-checking tasks and the use of AI technologies to counter disinformation.

AI technologies, particularly NLP (Natural Language Processing) and automatic fact-checking, demonstrate a high level of effectiveness in detecting fake information. Models such as BERT and GPT-4 analyze semantic patterns and sentence structures, achieving accuracy levels of up to 95% when integrated with SVM systems [6]. Computer vision is also actively utilized in combating deepfakes. Tools such as Sensity AI employ video forensics to detect artifacts like inconsistent blinking or anatomical errors. These technologies help to analyze the spread of fake content on social networks by examining geographical and temporal patterns [7].

It is also important to highlight hybrid human-machine systems, which combine AI algorithms with expert verification. The UK's Full Fact system integrates NLP models with human fact-checking of political claims [2]. In Ukraine, a similar approach is employed by the Trusted Flaggers organization.

In the research [5], a model of an information warfare system for the National Security and Defense Coordination Center has been developed. According to the authors, this system should include a monitoring platform integrating text classification algorithms to analyze disinformation trends. Creating an open annotated database of fake content, similar to EUvsDisinfo, is considered crucial. Decentralized systems, particularly blockchain-based solutions, could form the foundation for independent fact-checking databases. Predictive algorithms, such as MIT's Veracity, are also effective, anticipating the virality of fake content several hours before it spreads widely [3].

For Ukraine, which has been subjected to an ongoing information war for many years, the implementation of AI tools for detecting disinformation is, according to the authors, a matter of national security. To effectively counter disinformation, it is necessary to adopt a comprehensive approach that actively integrates various artificial intelligence technologies, tools, and platforms.

References

1. RAND: Towards an AI-Based Counter-Disinformation Framework: Website. URL: www.rand.org/pubs/commentary/2021/03/towards-an-ai-based-counter-disinformation-framework.html (Accessed 04.04.2025).

2. Frontiers: The use of artificial intelligence in counter-disinformation: Website. URL: www.frontiersin.org/journals/political-science/articles/10.3389/fpos.2025.1517726/full (Accessed 04.04.2025).

3. Sedova K., McNeill Ch., Johnson A., Joshi A., Wulkan I. AI and the Future of Disinformation Campaigns : CSET Policy Brief. 2021. URL: https://cset.georgetown.edu/wp-content/uploads/CSET-AI-and-the-Future-of-Disinformation-Campaigns-Part-2.pdf (Accessed 04.04.2025).

4. Доренський О.П. Модель поведінки держави в умовах проявів ознак інформаційної експансії, агресії, війни. *Інформаційна безпека держави, суспільства та особистості*. 2015. С. 131-133.

5. Доренський О.П., Улічев О.С., Задорожний К.О., Коваленко А.С., Дрєєва Г.М. Концептуальна модель системи інформаційного протиборства координаційного центру з питань національної безпеки і оборони. *Центральноукраїнський науковий вісник. Технічні науки*. 2024. Вип. 10(41). Ч. 2. С. 23-31.

6. Ashiq Mohammed M., Pradeesh E., Jeevanantham M., Anandhu B.S., Fake News Classification Using Machine Learning. *International Journal of Engineering Research & Technology (IJERT)*. 2023. Volume 11, Issue 03. DOI : 10.17577/IJERTCONV11IS03020 (Accessed 03.04.2025).

7. The Trusted Web: 13 AI-Powered Tools for Fighting Fake News: Website. URL: https://thetrustedweb.org/ai-powered-tools-for-fighting-fake-news/ (Accessed 03.04.2025).

УДК 338.47: 65.012.8

Ю. А. Невмержицька, *здобувач вищої освіти, 1-й курс,*
Науковий керівник: Ю.В. Білявська, *доцент кафедри менеджменту, к. е. н., доцент*
yuliianevmerzicka@gmail.com, y/biliavska@knute.edu.ua
Державний торговельно-економічний університет, Київ

МЕНЕДЖМЕНТ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В СУЧАСНОМУ ЦИФРОВОМУ СЕРЕДОВИЩІ

Система менеджменту інформаційної безпеки є невід'ємною частиною загального управління організацією та спрямована на забезпечення надійного захисту даних. Вона охоплює створення, впровадження, контроль, аналіз і вдосконалення заходів безпеки з урахуванням усіх потенційних загроз і ризиків. До її складових належать організаційна структура, політики, стратегічне планування, регламентовані процедури, робочі процеси та необхідні ресурси. Важливу роль у її ефективності відіграє дотримання встановлених правил усіма працівниками, адже навіть найкращі заходи безпеки не будуть ефективними без належної дисципліни та відповідальності співробітників.

Інформаційна безпека у сучасному цифровому середовищі відіграє надзвичайно важливу роль, оскільки з розвитком технологій зростає кількість загроз, пов'язаних із витоком даних, кібератаками та шахрайством. У сучасному світі інформація стала одним із найцінніших ресурсів, і її захист є критично необхідним як для окремих користувачів, так і для бізнесу, державних установ та цілих країн. Інформаційна безпека в глобальних процесах набуває особливого значення і, внаслідок її тісних взаємовпливів з економічною та національною безпекою, вносить свій значний вклад у глобальну безпеку[1].

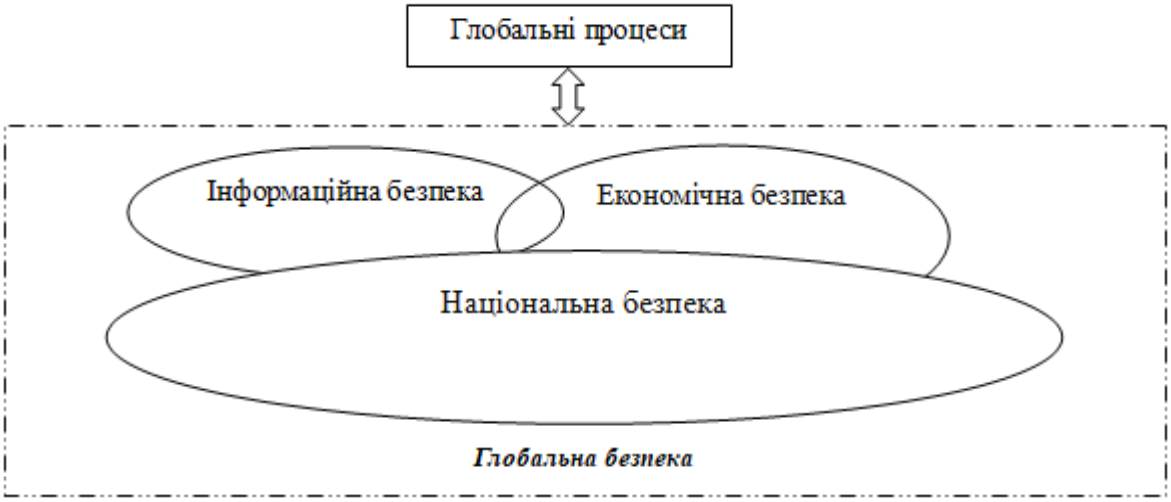


Рис. 1 Роль та місце інформаційної безпеки у процесі глобалізації

Система управління інформаційною безпекою є складовою загальної системи управління організацією і виконує важливу функцію забезпечення захисту інформації. Вона охоплює весь цикл заходів безпеки – від створення та впровадження до постійного моніторингу, аналізу та вдосконалення, з урахуванням усіх можливих ризиків. Система менеджменту інформаційної безпеки включає:

- планування заходів та розробка комплексної системи захисту інформації– визначення стратегії безпеки, оцінка загроз та розробка ефективних механізмів захисту даних;
- організація роботи щодо захисту інформації – створення спеціального підрозділу для контролю безпеки, розробка чітких правил, політик та стратегій, що визначають порядок захисту інформації. Впровадження наукових принципів, таких як дотримання законодавчих вимог, економічна доцільність, відповідальність, зв'язок теорії з практикою, професійна підготовка персоналу та забезпечення балансу між відкритістю інформації і необхідним рівнем конфіденційності;
- мотивація персоналу – створення умов для підвищення відповідальності співробітників у питаннях інформаційної безпеки. Працівники можуть як захищати, так і несвідомо чи свідомо загрожувати стабільності організації, тому важливо впроваджувати психологічні та організаційні заходи для підвищення рівня кібергігієни;
- контроль і аудит безпеки – регулярні перевірки ефективності впроваджених заходів, аналіз дотримання політики безпеки та коригування стратегії відповідно до сучасних загроз.

Таким чином, ефективне управління інформаційною безпекою є ключовим елементом стабільної роботи підприємства, що дозволяє мінімізувати ризики, підвищити довіру до компанії та забезпечити безперебійну роботу всіх її систем[2].

Важливою проблемою забезпечення інформаційної безпеки бізнесу є також застосування розгалуженої системи економічних заходів. Перш за все, варто усвідомлювати, що проблема захисту інформації має витратний аспект, який необхідно враховувати, порівнюючи позитивний ефект від захищеності інформаційних ресурсів і розмір витрат, які мають бути понесені на забезпечення такого захисту. Крім того, реалізація окремих заходів має проводитися на підставі співставлення вигід від захисту інформації і можливих втрат, які можуть бути понесені в результаті відсутності такого захисту. Цілком зрозуміло, що в разі відсутності економічної доцільності окремі заходи щодо захисту інформації не варто реалізовувати[3].

В умовах цифрової реальності питання менеджменту інформаційної безпеки стає особливо актуальним для всіх суб'єктів господарювання, зокрема для малого бізнесу. Обмежені фінансові та кадрові ресурси ускладнюють створення повноцінної системи захисту даних, що включає спеціалізовані підрозділи та комплексні заходи безпеки. У такій ситуації ефективним рішенням може стати використання зовнішніх постачальників послуг із кібербезпеки, які на основі субконтрактної системи беруть на себе виконання окремих функцій із забезпечення безпеки інформації.

З огляду на це, ключовим аспектом управління інформаційною безпекою є стратегічне планування та формування спеціалізованого бюджету. Виділення окремих фінансових ресурсів на кібербезпеку дозволяє підприємствам ефективно управляти ризиками, підвищувати рівень захищеності та адаптуватися до сучасних загроз. Грамотне управління інформаційною безпекою, навіть із залученням сторонніх компаній, стає необхідною умовою стабільного розвитку бізнесу в цифрову епоху[4].

Інформаційна безпека в умовах цифрової трансформації України (розвитку інформаційного суспільства) – це система суспільних відносин, спрямованих на створення та підтримання належного рівня захисту автоматизованих (комп'ютеризованих) інформаційних систем і телекомунікаційних мереж. Вона охоплює комплекс організаційних, правових і технологічних (технічних та програмних) заходів, які забезпечують збереження, захист, а також запобігання та нейтралізацію загроз природного, техногенного чи соціального характеру. Недотримання цих заходів може призвести до порушення або зупинки функціонування відповідних інформаційних систем, що впливає на стабільність соціотехнічного середовища[5].

Менеджмент інформаційної діяльності передбачає застосування економічних методів, які сприяють ефективному управлінню інформаційними ресурсами та забезпеченню їх захисту. Зокрема, важливо впроваджувати мотиваційні механізми, що стимулюють працівників до відповідального ставлення до обробки даних, підвищення рівня цифрової грамотності та дотримання політик безпеки. Водночас необхідно використовувати заходи, які запобігають діям, що можуть спричинити витік конфіденційної інформації, порушення цілісності баз даних чи негативно вплинути на репутацію організації.

Загалом, ефективне управління інформаційною діяльністю в умовах цифрової економіки потребує комплексного підходу, який об'єднує організаційні, технічні та економічні методи. Взаємодія цих складових дозволяє мінімізувати ризики, що виникають у процесі цифрової трансформації, і сприяє стабільному розвитку бізнесу та його інформаційних систем[6].

Список літератури:

1. Менеджмент інформаційної безпеки: Навчальний посібник / Тардаскіна Т.М., Кононович В.Г. - Одеса: ОНАЗ ім. О.С. Попова, 2009. - 265с.
2. МЕНЕДЖМЕНТ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ. URL: http://dspace.wunu.edu.ua /bitstream/316497/36025/1/Менеджмент_ІБ_Якименко_лекції.pdf.
3. ІНФОРМАЦІЙНА БЕЗПЕКА БІЗНЕСУ В УМОВАХ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ ЕКОНОМІКИ. URL: https://ir.kneu.edu.ua/server/api/core/bitstreams/1cc3d220-2a29-47cc-ba4b-e2cbca7cb870/content.
4. Шопіна Ірина Миколаївна. ІНФОРМАЦІЙНА БЕЗПЕКА ЦИФРОВОЇ ТРАНСФОРМАЦІЇ. *НАУКОВИЙ ВІСНИК 1'2023*. URL: https://dspace.lvduvs.edu.ua/bitstream /1234567890/5636/1/4.pdf.
5. Забезпечення інформаційної безпеки цифрових програмно керованих АТС Інформаційна безпека телефонного зв'язку: навч. посібник / [Кононович В.Г., Стайкуца С.В., Тардаскіна Т.М., Шинкарчук Т.М.] За ред. чл.-кор. МАЗ В.Г. Кононовича. – Одеса: ОНАЗ ім. О.С. Попова, 2010. – С. 168.
6. ІНФОРМАЦІЙНА БЕЗПЕКА БІЗНЕСУ В УМОВАХ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ ЕКОНОМІКИ. URL: https://ir.kneu.edu.ua/server/api/core/bitstreams/1cc3d220-2a29-47cc-ba4b-e2cbca7cb870/content.

УДК 004.056.5

В.А. Резніченко, *викладач*, А.Я. Клюй, *студенка 2-го курсу*
e-mail: upsbilly@meta.ua, nastya.klui@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

МЕТОДИ ФІЛЬТРАЦІЇ КОНТЕНТУ ЯК ЗАСІБ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ: ОГЛЯД І ОЦІНКА ЕФЕКТИВНОСТІ

Фільтрація контенту — важливий інструмент захисту інформації у комп'ютерних системах та мережах.

Завдяки їй можна оперативно виявляти й блокувати загрози, такі як віруси, фішингові атаки та інші небажані дані. Шкідливі інформаційні повідомлення активно поширюються, особливо під час збройної агресії росії проти України. За даними звіту CYBER DIGEST (2024 р.) понад 60% користувачів регулярно стикаються з онлайн-загрозами [3]. Це свідчить про реальну потребу впровадження сучасних методів контент-фільтрації.

Попри наявність багатьох підходів, ефективність кожного методу залежить від типу мережі та характеру загроз. Внаслідок цього виникає невизначеність у виборі оптимального рішення. Тому метою цього аналізу є огляд основних методів фільтрації контенту, порівняння їх переваг і недоліків, а також оцінка здатності захищати інформаційні системи та мережі від актуальних кіберзагроз.

Огляд ґрунтується на методах, що найчастіше згадуються у наукових джерелах:

1. **Чорні списки URL.** За словами Дж.Р. Андерсона [1], блокування доступу до відомих шкідливих сайтів є простим і швидким способом запобігти поширенню загроз на мережевому рівні. Цей метод вимагає мінімум ресурсів, але не здатен виявляти нові загрози та потребує регулярного оновлення бази.

2. **Ключові слова**. Д.О. Бухаленков і Т.М. Заболотна [2] зазначають, що аналіз тексту на наявність заборонених чи ризикованих слів широко використовується у спам-фільтрах і системах модерації. Цей підхід швидко знаходить небезпечний контент. Проте часто призводить до великої кількості хибних спрацьовувань, оскільки ті самі слова можуть мати різне значення.

3. **Машинне навчання**. Як підкреслює Т. Мітчелл [5], алгоритми, що навчаються на великих обсягах даних, підвищують точність фільтрації та дозволяють системам швидко адаптуватися до нових загроз. Однак цей підхід вимагає значних обчислювальних ресурсів і створення якісних тренувальних вибірок.

4. **Контекстний аналіз**. За даними К. Дінакара, Г. Лібермана і Р. Пікарда [4], семантичний розбір повідомлень дозволяє точніше виявляти приховані загрози та знижувати кількість помилкових блокувань. Цей метод базується на сучасних моделей обробки мови (GPT, BERT тощо) і потребує високих обчислювальних потужностей.

З урахуванням проаналізованих джерел [1–5] було сформовано власну узагальнену порівняльну таблицю методів фільтрації контенту (Таблиця 1) з позиції їхньої здатності підвищувати рівень безпеки в комп'ютерних системах та мережах. У ній відображено принцип роботи кожного методу, його ключові переваги й недоліки, а також розглядається сфера застосування. Такий підхід дає змогу порівняти ефективність різних інструментів фільтрації та обрати оптимальну стратегію захисту інформації від сучасних кіберзагроз.

Таблиця 1

Огляд методів фільтрації контенту з погляду забезпечення інформаційної безпеки

Метод	**Принцип роботи**	**Переваги**	**Недоліки**	**Сфера застосування**	**Сфера застосування**
Чорні списки URL	Перевірка звернень до відомих шкідливих сайтів	Швидка реакція, низькі вимоги до ресурсів	Не бачить нові чи невідомі загрози, потребує оновлень	Мережевий рівень, корпоративні проксі, базовий рівень безпеки.	Мережеві фільтри, проксі, базовий захист
Ключові слова	Пошук у тексті заборонених слів	Швидке блокування типових загроз, простота реалізації.	Хибнопозитивні спрацьовування, бо відсутній контекст	Модерація контенту, антиспам-системи	Модерація, антиспам, батьківський контроль
Машинне навчання	Використання алгоритмів машинного навчання	Висока точність, здатність адаптуватися до нових загроз	Вимагає значних обчислювальних ресурсів та якісних тренувальних даних	Антиспам, аналіз мережевого трафіку, виявлення аномалій	Антифішинг, виявлення аномалій, кіберзахист
Контекстний аналіз	Семантичний розбір тексту з урахуванням контексту	Точніше виявлення прихованих загроз, хибнопозитивів менше	Високі вимоги до обчислювальних потужностей, складна інтеграція	Моніторинг соцмереж, боротьба з дезінформацією	Соцмережі, дезінформація, глибока фільтрація

Розширений опис у контексті Таблиці 1 свідчить, що кожен із розглянутих методів має як сильні, так і слабкі сторони. Як видно, кожен критерій грає ключову роль в оцінці ефективності контент-фільтрації. Наприклад, методи *чорні списки URL* та *фільтрація за ключовими словами* відзначаються простотою впровадження та низькими вимогами до ресурсів, проте їм бракує гнучкості й здатності оперативно реагувати на нові загрози. Навпаки, методи, що базуються на *машинному навчанні* та *контекстному аналізі,* демонструють високу точність, проте вимагають значних обчислювальних потужностей і складного налаштування.

З огляду на наведене, жоден із розглянутих методів не є універсальним. Проте комплексне застосування методів, тобто багаторівневий підхід, дозволяє поєднати їх переваги та мінімізувати недоліки. Логіка цього підходу полягає в поступовому ускладненні аналізу: кожен рівень доповнює та підсилює результати попереднього, одночасно знижуючи загальне навантаження на систему:

1. Перший рівень — чорні списки URL: забезпечує базове, швидке блокування відомих загроз за мінімальних витрат ресурсів.

2. Другий рівень — фільтрація за ключовими словами: здійснює попередній аналіз текстового контенту для оперативного виявлення стандартних загроз, незважаючи на можливі хибнопозитиви.

3. Третій рівень — машинне навчання: дозволяє здійснювати глибший аналіз, розпізнає приховані шаблони в даних та адаптується до нових типів загроз.

4. Четвертий рівень — контекстний аналіз: застосовує семантичний і лінгвістичний розбір тексту, що дозволяє враховувати контекст повідомлень та точніше ідентифікувати загрози, мінімізуючи хибнопозитиви.

Висновки. У роботі здійснено огляд методів фільтрації контенту та оцінено їх ефективність за ключовими критеріями в комп'ютерних науках. Отримані результати підтверджують доцільність застосування багаторівневого підходу для забезпечення інформаційної безпеки. Завдяки поступовому ускладненню аналізу – від базового блокування («чорні списки URL», «фільтрація за ключовими словами») до глибокого аналізу («машинне навчання», «контекстний аналіз») – система стає здатною адаптуватися до нових типів контенту й загроз. Таким чином, комплексне використання зазначених методів є важливим інструментом для підвищення безпеки інформаційних систем і вдосконалення процесів обробки даних.

Список літератури

1. Anderson R.J. Security Engineering: A Guide to Building Dependable Distributed Systems / R.J. Anderson. 3rd ed. Hoboken: John Wiley & Sons, 2020, 1232 p. URL: https://www.cl.cam.ac.uk/archive/rja14/book.html?utm_source=chatgpt.com (дата звернення 09.04.2025)

2. Бухаленков Д.О., Заболотня Т.М. Модифікований метод пошуку ключових слів та термінів у текстових даних. Проблеми програмування, 2024, №1, с. 12–22. URL: file:///Users/masya/Downloads/602-1429-1-SM.pdf (дата звернення 08.04.2025)

3. CYBER DIGEST: Огляд подій у сфері кібербезпеки. Січень, 2024, 44 с. URL: https://www.rnbo.gov.ua/files/2024/NATIONAL_CYBER_SCC/Cyber%20digest/Cyber%20digest_Jan_2024_UA.pdf (дата звернення 09.04.2025)

4. Dinakar K., Lieberman H., Picard R. «Using Common Sense Reasoning to Detect and Respond to Cyberbullying» // ACM Transactions on Interactive Intelligent Systems, 2022, Vol. 2, № 3, Article 18, p. 1-27.

5. Mitchell T.M. «Machine Learning» / T.M. Mitchell. New York: McGraw-Hill, 1997, 432 p. URL: https://www.cs.cmu.edu/~tom/files/MachineLearningTomMitchell.pdf?utm_source=chatgpt.com (дата звернення 09.04.2025)

УДК 004.93:316.77

В.Р. Карабчук, *аспірант 2 курсу*
dedeatbomb@gmail.com
Науковий керівник: канд. техн. наук О.С. Улічев
ПВНЗ «Європейський університет», Київ

МЕТОДИ ВИЯВЛЕННЯ ТА БОРОТЬБИ З ДЕЗІНФОРМАЦІЄЮ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ

У сучасному інформаційному середовищі дезінформація перетворилась на серйозну загрозу для суспільства, політичної стабільності та національної безпеки. Завдяки швидкому розвитку цифрових технологій та соціальних мереж фейкові новини, маніпулятивні матеріали і зображення, змінені за допомогою штучного інтелекту, можуть поширюватися з великою швидкістю і охоплювати широкі аудиторії [1, 4, 9]. Такий контент часто має на меті викликати недовіру до державних інституцій, дестабілізувати суспільну думку або вплинути на демократичні процеси.

Як зазначають Дмитроца і Дацик [1], платформи на кшталт Facebook стали одним із головних каналів розповсюдження дезінформації, що підвищує вимоги до ефективних систем контролю за інформаційним простором. При цьому традиційні методи перевірки інформації, зокрема ручна фактчекінг-діяльність, демонструють обмежену ефективність в умовах великих обсягів даних і швидкої динаміки їх розповсюдження [2, 6].

У відповідь на ці виклики дедалі ширше впроваджуються інструменти штучного інтелекту. Як показують дослідження Дзюбановської [3] та Легомінової і Тищенка [4], алгоритми обробки природної мови, глибокого навчання і нейронних мереж демонструють високу точність у задачах розпізнавання фейкової інформації, аналізу джерел та виявлення бот-активності.

Отже, проблема боротьби з дезінформацією вимагає застосування комплексного технологічного підходу. Використання можливостей ШІ дозволяє реалізувати масштабовані рішення, що здатні ефективно виявляти неправдиву інформацію, перевіряти факти в автоматичному режимі та ідентифікувати мережі поширення контенту сумнівного походження [5, 7].

Основна мета дослідження

Метою дослідження є комплексний аналіз сучасних інструментів і технологій штучного інтелекту, які використовуються для виявлення, класифікації та нейтралізації дезінформації, а також визначення шляхів їх ефективного практичного впровадження в інформаційні системи, соціальні медіа та журналістику. У межах дослідження також розглядаються архітектури моделей машинного навчання, рівень їх точності, застосовність у реальному часі, а також проблеми етики, довіри до ШІ та його прозорості.

Основні технології:

Обробка природної мови (NLP)

Сучасні мовні моделі (BERT, RoBERTa, GPT) дозволяють аналізувати стиль, зміст та контекст повідомлень для виявлення маніпуляцій [3, 4]. Такі моделі можуть досягати точності понад 85% у задачах класифікації контенту як достовірного або фейкового [10].

Генеративні змагальні мережі (GANs)

Генеративні змагальні мережі активно використовуються для створення deepfake-контенту — фотореалістичних відео та зображень, які важко відрізнити від справжніх [4, 7]. Такі матеріали становлять серйозну загрозу в контексті дезінформації, оскільки можуть маніпулювати громадською думкою або дискредитувати осіб. Для виявлення deepfake-зображень найчастіше застосовуються згорткові нейронні мережі (CNN), що аналізують мікродеталі візуального контенту: артефакти рендерингу, спотворення текстури шкіри, неприродні тіні, а також інші неузгодженості, малопомітні для людського ока [5].

Автоматизована перевірка фактів

Системи як ClaimBuster або FullFact AI здійснюють автоматичне порівняння тверджень із базами достовірної інформації в реальному часі [6]. Такі системи вже впроваджуються у великих медіа-агентствах та соціальних мережах [7].

Аналіз поширення інформації в соцмережах

Graph Neural Networks (GNN) використовуються для аналізу мережевих структур: взаємодії користувачів, поширення контенту, активності ботів [1, 8]. Ці моделі дозволяють виявляти аномальні схеми розповсюдження дезінформації.

Етичні аспекти

Застосування ШІ в медіа потребує етичного підходу. Системи мають бути прозорими, інтерпретованими та недискримінаційними [7]. Важливо забезпечити права користувачів, уникати цензури та зловживань [8].

Висновки

Технології штучного інтелекту відіграють усе більш важливу роль у виявленні та протидії дезінформації, що активно поширюється в цифровому середовищі. Серед ключових напрямів застосування варто виділити моделі обробки природної мови, які дають змогу аналізувати зміст текстів і виявляти маніпулятивні або неправдиві твердження; системи візуального аналізу, здатні розпізнавати підроблені зображення та відео; а також сервіси автоматизованої перевірки фактів та інструменти, що досліджують динаміку поширення контенту у соціальних мережах. Подальші дослідження мають бути спрямовані на об'єднання зазначених підходів у єдині платформи, які зможуть ефективно реагувати на складні форми дезінформації. Особливу увагу слід приділяти етичним аспектам впровадження таких технологій, зокрема питанням прозорості алгоритмів, відкритості моделей та недопущення їх використання у спосіб, що порушує права людини або свободу слова.

Список літератури

1. Дмитроца Л.П., Дацик С.В. Застосування методів штучного інтелекту для виявлення та протидії дезінформації у Facebook // Матеріали конференції IMSTT. 2023. С. 37–38.

2. Макарова М. Інструменти сприяння інформаційній безпеці в медіапросторі // Вісник Книжкової палати. 2024. №8. С. 23–30.

3. Дзюбановська Н.В. Штучний інтелект для виявлення фейкової інформації: міжнародні тренди і можливості імплементації // Інноваційна економіка. 2024. №3.

4. Легомінова С.В., Тищенко В.С. та ін. Штучний інтелект та соціальні мережі: підходи до виявлення фейкової інформації // Телекомунікаційні та інформаційні технології. 2024. №4.

5. Лаптєва Т.О. Методи виявлення дезінформації на основі експертних оцінок: дис. ... канд. техн. наук. Харків: НТУ «ХПІ», 2025.

6. Дмитроца Л.П., Дацик С.В. Аналіз інструментів штучного інтелекту для виявлення дезінформації в новинах Facebook // Матеріали конференції IMSTT. 2023. С. 35–36.

7. Лауер Т. Штучний інтелект і дезінформація: виклики регулювання // Digital Communication. 2024.

8. Флорес А. Штучний інтелект і дезінформація: можливості та ризики в умовах війни // Укрінформ. 2023.

9. Семенов І.М. Соціальні мережі та дезінформація: виклики сучасності // Соціальні комунікації. 2019. №2. С. 88–95.

10. Ковальчук О.В. Методи машинного навчання у виявленні фейкових новин // Журнал інформаційних технологій. 2020. №3. С. 45–52.

УДК 004.056

Д.М. Чікін[1], С.В. Науменко[2,3], І.О. Розломій[1]
d.m.chikin.fitis23@chdtu.edu.ua, naumenko.serhii1122@vu.cdu.edu.ua, inna-roz@ukr.net
[1]*Черкаський державний технологічний університет, Черкаси*
[2]*Черкаський національний університет імені Богдана Хмельницького, Черкаси*
[3]*Active Bridge, Черкаси*

ЗАХИСТ ПЕРСОНАЛЬНИХ ДАНИХ В IOT-ПРИСТРОЯХ ІЗ ЗАСТОСУВАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Розвиток Інтернету речей (IoT) супроводжується активним впровадженням штучного інтелекту (ШІ), що підвищує ефективність пристроїв та їхню здатність до самостійного аналізу інформації. Одним з ключових викликів є захист персональних даних, оскільки велика кількість розумних пристроїв здійснює обробку чутливої інформації користувачів. Сучасні дослідження вказують на необхідність впровадження багаторівневих механізмів безпеки, які поєднують криптографічні методи, політики контролю доступу та аналіз поведінкових характеристик користувачів [1].

Використання штучного інтелекту у сфері безпеки IoT-пристроїв дозволяє підвищити рівень захисту даних за рахунок адаптивного аналізу загроз, автоматичної ідентифікації аномальних активностей та оптимізації процедур автентифікації [2]. Впровадження методів машинного навчання сприяє покращенню ефективності системи виявлення атак шляхом побудови моделей поведінки користувачів та аналізу відхилень від нормативних патернів. Крім того, сучасні алгоритми можуть забезпечити динамічне налаштування політик безпеки залежно від контексту використання пристрою та рівня ризику компрометації персональних даних.

Одним із найбільш перспективних напрямів розвитку є впровадження гомоморфного шифрування для захисту переданих даних без необхідності їхнього розшифрування на проміжних вузлах мережі. Це дозволяє значно знизити ймовірність несанкціонованого доступу до конфіденційної інформації та забезпечити захищене виконання обчислювальних операцій у розподілених середовищах. Додатково, комбінування гомоморфного шифрування із захищеними багаторазовими автентифікаційними механізмами підвищує загальний рівень безпеки екосистеми IoT.

На рис. 1 представлено узагальнену схему механізмів захисту даних в IoT-пристроях із використанням штучного інтелекту.

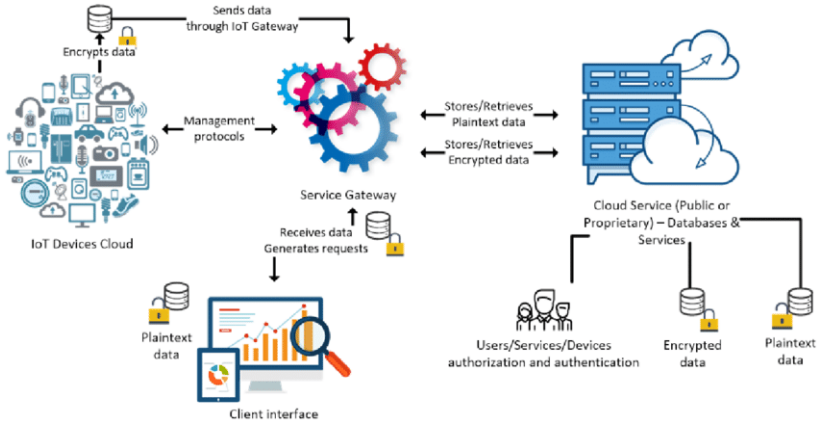


Рис. 1. Основні механізми захисту даних у IoT-пристроях [5]

Додатково, представлено основні методи застосування штучного інтелекту для підвищення безпеки IoT-пристроїв (табл. 1).

Після аналізу основних методів застосування штучного інтелекту в безпеці IoT-пристроїв можна зробити висновок, що ці технології не тільки підвищують рівень захисту, а й дозволяють значно автоматизувати процеси виявлення загроз та управління доступом. Інтелектуальні алгоритми на основі машинного навчання дають змогу системам розпізнавати невідомі атаки, аналізуючи поведінкові аномалії користувачів і автоматично змінюючи параметри безпеки в реальному часі. Впровадження адаптивних систем дозволяє значно скоротити час реагування на потенційні загрози та зменшити ймовірність несанкціонованого доступу до персональних даних.

Сучасні інформаційні системи, що обслуговують Інтернет речей, стикаються з численними викликами у сфері безпеки. Динамічне середовище, велика кількість пристроїв та їхня розподілена структура ускладнюють реалізацію статичних моделей контролю доступу. Саме тому необхідно впроваджувати підходи, які можуть адаптуватися до нових умов і загроз у реальному часі, забезпечуючи при цьому ефективний рівень захисту. Використання штучного інтелекту відкриває можливості для вдосконалення процесів оцінки ризиків та

управління доступом, дозволяючи впроваджувати моделі, що здатні навчатися та прогнозувати небезпеки на основі отриманих даних.

Таблиця 1

Методи застосування ШІ в IoT

Метод	Опис
Автоматизоване виявлення загроз	Використання алгоритмів машинного навчання для виявлення атак та аналізу вторгнень
Аналіз поведінкових аномалій	Моніторинг активності користувача для виявлення підозрілих дій, таких як аномальний трафік або несанкціоновані входи
Інтелектуальне управління доступом	Динамічне налаштування рівня доступу до ресурсів залежно від контексту, поведінки користувача та рівня ризику
Захист на основі блокчейну	Використання децентралізованих технологій для безпечного зберігання даних і запобігання їх підробці
Квантова криптографія	Новітній метод шифрування, який гарантує захист від атак навіть із використанням квантових обчислень

Підходом, що забезпечує таку адаптивність, є застосування математичної моделі на основі Марковських процесів прийняття рішень (MDP). Визначимо стан системи як, множину можливих дій, ймовірність переходу між станами та функцію винагороди. Мета полягає в знаходженні оптимальної політики, яка максимізує очікувану кумулятивну винагороду:

$$V^{\pi}(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s, \pi]$$

де s – стан системи, що може відповідати рівню ризику компрометації пристрою, а – множина можливих дій, тобто заходи безпеки, які можуть бути застосовані, $R(s_t, a_t)$ – функція винагороди, яка враховує як ефективність захисту, так і витрати ресурсів, γ – коефіцієнт дисконтування, що враховує майбутні винагороди, $V^{\pi}(s)$ – очікувана кумулятивна винагорода при дотриманні стратегії.

У цьому контексті стан може відповідати рівню ризику компрометації пристрою, діями можуть бути заходи безпеки, а винагорода враховує як ефективність захисту, так і витрати ресурсів. Оптимізація політики може здійснюватися за допомогою методів глибокого підкріпленого навчання, що дозволяє системі адаптуватися до динамічних загроз.

Запропонована модель органічно доповнить розділ, у якому розглядаються механізми адаптивного управління доступом. Оптимальним місцем для її розміщення є частина, що слідує за описом методів використання штучного інтелекту у сфері інформаційної безпеки. Це дасть змогу обґрунтувати математичний підхід до проблеми та показати його застосування у практичних сценаріях.

Одним із ключових напрямів розвитку є інтеграція штучного інтелекту з технологіями розподілених реєстрів, такими як блокчейн. Це забезпечує високий рівень довіри до збереження персональних даних у мережах IoT та унеможливлює їх підробку або несанкціоновану модифікацію. Додатково, використання квантових алгоритмів шифрування у перспективі може суттєво покращити стійкість до атак з боку зловмисників, що застосовують сучасні методи криптоаналізу. У комплексі ці рішення дозволяють створити стійку систему захисту, яка відповідатиме сучасним викликам та нормативним вимогам у сфері інформаційної безпеки.

Окрему увагу необхідно приділити правовим аспектам захисту персональних даних, оскільки нормативна база у сфері інформаційної безпеки змінюється відповідно до новітніх викликів [3]. Важливими є стандарти, що регулюють обробку даних у IoT, такі як ISO/IEC 27001 та GDPR, які визначають ключові вимоги до забезпечення конфіденційності та неперервного моніторингу ризиків. Комплексний підхід, що включає правові, технічні та організаційні заходи, дозволяє значно підвищити рівень безпеки персональних даних у розумних пристроях.

Список літератури

1. Shafagh H., Hithnawi A., Fereidooni H., Mühlebach M., Karame G. Poster: Towards blockchain-based auditable storage and sharing of IoT data // Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. – 2017. – С. 2541-2543.

2. Розломій, І.О., Фауре, Е.В., & Науменко, С.В. (2025). Методи аутентифікації у вбудованих системах з обмеженими обчислювальними ресурсами. Measuring and computing devices in technological processes, (1), 29-35. https://doi.org/10.31891/2219-9365-2025-81-4

3. Hussein, R., Wurhofer, D., Strumegger, E. M., Stainer-Hochgatterer, A., Kulnik, S. T., Crutzen, R., & Niebauer, J. (2022). General data protection regulation (GDPR) toolkit for digital health. MEDINFO 2021: One World, One Health–Global Partnership for Digital Innovation, 222-226.

УДК 004.056

К.В. Ротань[1], І.О. Розломій[1]
k.v.rotan.fitis23@chdtu.edu.ua, inna-roz@ukr.net
[1]*Черкаський державний технологічний університет, Черкаси*

ПОРІВНЯННЯ АЛГОРИТМІВ ASCON ТА ELEPHANT У КОНТЕКСТІ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ВБУДОВАНИХ ПРИСТРОЇВ

У сучасних умовах стрімкого розвитку Інтернету речей та збільшення кількості вбудованих пристроїв з обмеженими обчислювальними ресурсами зростає потреба у використанні ефективних і водночас надійних механізмів криптографічного захисту. Традиційні криптографічні алгоритми, розроблені для систем із високою продуктивністю, часто не відповідають вимогам енергоефективності, розміру коду та апаратної реалізації, які є критичними у сфері вбудованих систем [1]. Саме тому актуальним є дослідження полегшених криптографічних алгоритмів, здатних забезпечити баланс між безпекою та ресурсоспоживанням.

Серед таких алгоритмів особливу увагу привертають ASCON, рекомендований як стандарт від NIST для легковагового шифрування, та ELEPHANT, який демонструє високу ефективність у середовищах з обмеженою пам'яттю та обчислювальними потужностями. Обидва алгоритми орієнтовані на автентифіковане шифрування та забезпечують захист цілісності й конфіденційності даних у режимі end-to-end.

ASCON і Elephant – це блочні шифри, які використовуються для полегшеного криптографічного захисту, особливо в умовах обмежених ресурсів, таких як IoT та вбудовані системи. Обидва алгоритми базуються на перестановочних функціях, але мають суттєві відмінності в архітектурі, рівні безпеки та продуктивності.

ASCON використовує мережу підстановки-перестановки (SPN) та базується на режимі MonkeyDuplex, що забезпечує високу криптографічну стійкість. У нього є дві основні варіації: ASCON-128 із 64-бітним блоком і ASCON-128a, що працює з 128-бітним блоком. Крім того, варіант ASCON-80pq забезпечує стійкість до квантових атак. Його функції хешування ASCON-HASH і ASCON-HASHA використовують губковий (sponge) режим роботи. Завдяки 5-бітним S-блокам і 64 паралельним операціям ASCON демонструє високу ефективність як у програмному, так і в апаратному середовищі.

Загальна схема алгоритму AEAD (Authenticated Encryption with Associated Data) для ASCON зображена на рис. 1. Ця схема демонструє етапи ініціалізації, обробки асоційованих даних, шифрування (або дешифрування) відкритого тексту та фіналізації з генеруванням тега автентифікації.

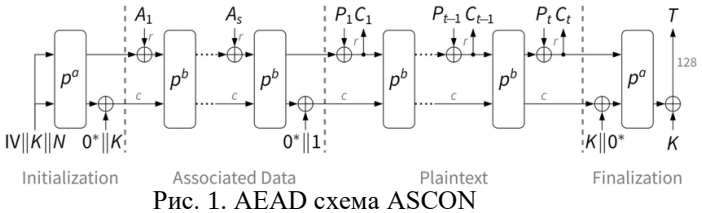


Рис. 1. AEAD схема ASCON

На основі представленої схеми видно, що ASCON проходить три основні етапи: ініціалізацію, обробку асоційованих даних і шифрування або хешування. Під час ініціалізації змішуються ключ і одноразове значення (nonce), після чого кожен блок відкритого тексту проходить XOR-операцію з внутрішнім станом перед обчисленням перестановки.

На кожному раунді перестановки стан змінюється шляхом додавання константи, застосування нелінійного S-блоку (паралельно до кожного слова стану) та лінійного дифузійного шару. Саме така структура і забезпечує криптостійкість та ефективність. У хешуванні ASCON використовується класична схема губки (sponge construction), де повідомлення по частинах поглинається у внутрішній стан, а потім стискається для отримання хеш-значення. Детальна схема цієї операції наведена на рис. 2.

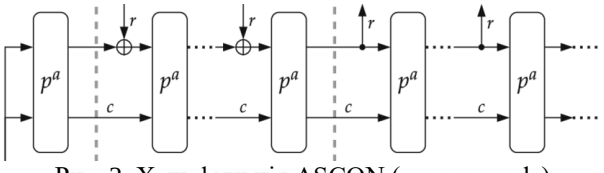


Рис. 2. Хеш-функція ASCON (sponge mode)

Алгоритм Elephant також є автентифікованим шифром на основі перестановок, розробленим Beyne та співавторами. Він використовує режим «спочатку шифрування, потім MAC» з підтримкою одноразового номера (nonce). В основі шифрування − режим лічильника, а для автентифікації − модифікована MAC-функція, заснована на перестановці та XOR-операціях. Процес шифрування Elephant показаний на рис. 3, де видно, як

кожен блок повідомлення шифрується за допомогою маски, згенерованої з перестановки, та одноразового номера.

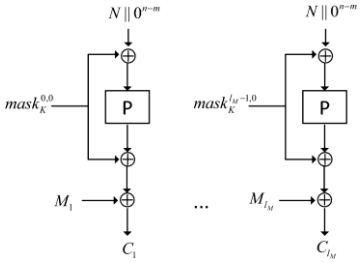


Рис. 3. Схема шифрування Elephant

Elephant, на відміну від ASCON, використовує режим лічильника (CTR) для шифрування і захищену MAC-функцію для автентифікації. Це алгоритм, оптимізований для паралельних обчислень, що робить його швидким у багатопотокових середовищах. Він має три варіанти: Dumbo, оптимізований для апаратного використання, Jumbo, що забезпечує 127-бітний рівень безпеки, і Delirium, який краще підходить для програмних реалізацій. На основі представленої схеми видно, що Elephant реалізує маскування даних перед їхньою обробкою, використовуючи LFSR для генерації масок. Це ускладнює атаки на основі аналізу споживаної потужності (SCA), адже кожен вихідний блок залежить не лише від ключа, а й від випадкових значень. Крім того, процедура автентифікації у Elephant заснована на MAC-функції, яка забезпечує перевірку цілісності даних після розшифрування.

Щодо стійкості до атак, ASCON більш захищений від атак на основі аналізу споживаної потужності (SCA) та атак із використанням помилок. Elephant використовує перестановки з LFSR, що ускладнює певні атаки, але через використання режиму CTR він може бути вразливим, якщо повторно використовуються одноразові номери (nonce). ASCON став стандартом у межах ініціативи NIST щодо полегшеної криптографії, тоді як Elephant залишився кандидатом і не був обраний. Це означає, що ASCON підходить для ширшого спектра застосувань, включаючи захист від квантових атак та оптимізацію для вбудованих пристроїв [2]. Elephant, у свою чергу, більше підходить для високопродуктивних мереж і сценаріїв, де важлива швидкість роботи. Якщо головним критерієм вибору є універсальність та безпека, ASCON є кращим варіантом. В таблиці 1 представлена порівняльна характеристика алгоритмів ASCON та Elephant.

Таблиця 1

Порівняльна характеристика ASCON та Elephant

Параметр	ASCON-128	ASCON-128a	ASCON-128a	Elephant (Dumbo)	Elephant (Jumbo)	Elephant (Delirium)
Ключ (біт)	128	128	80	128	128	128
Nonce (біт)	128	128	128	128	128	128
Розмір блоку (біт)	64	128	64	160	176	200
Розмір тегу (біт)	128	128	128	64	64	128

На основі даних, наведених у таблиці 1, можна зробити висновок, що ASCON-128 і його варіації демонструють кращий баланс між безпекою, ефективністю та відповідністю сучасним стандартам. Алгоритми сімейства Elephant мають більший розмір блоку та можуть бути корисними у сценаріях, де критичною є висока швидкість обробки в паралельних обчисленнях. Проте з точки зору криптостійкості, підтримки широкого спектра застосувань і схвалення в межах ініціативи NIST, саме ASCON є пріоритетним вибором для вбудованих систем з обмеженими ресурсами. Таким чином, при розробці захищених IoT-рішень доцільно надавати перевагу алгоритмам ASCON завдяки їхній гнучкості, надійності та високій сумісності з апаратними платформами.

Список літератури

1. Rozlomii, I., Symonyuk, V., Naumenko, S., & Mykhailovskyi, P. (2024). Модель безпеки взаємоповязаних обчислювальних пристроїв на основі полегшеної схеми шифрування для IoT. Computer-integrated technologies: education, science, production, (55), 191-198.

2. Kaur, J., Canto, A. C., Kermani, M. M., & Azarderakhsh, R. (2023). A comprehensive survey on the implementations, attacks, and countermeasures of the current NIST lightweight cryptography standard. arXiv preprint arXiv:2304.06222.

УДК 004.056

Д.В. Джирма[1], П.В. Михайловський[2,3], І.О. Розломій[1]
d.v.dzhyrma.fitis23@chdtu.edu.ua, mykhailovskyi.pavlo1123@vu.cdu.edu.ua, inna-roz@ukr.net
[1]*Черкаський державний технологічний університет, Черкаси*
[2]*Черкаський національний університет імені Богдана Хмельницького, Черкаси*
[3]*Active Bridge, Черкаси*

МЕТОДИ ВИЯВЛЕННЯ ТА ПРОТИДІЇ КІБЕРАТАКАМ НА ДЕРЖАВНІ ІНФОРМАЦІЙНІ СИСТЕМИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

У сучасних умовах стрімкої цифровізації всієї державної інфраструктури питання кібербезпеки набуває особливої актуальності. Державні інформаційні системи, які забезпечують функціонування критичних сервісів та обробку конфіденційної інформації, дедалі частіше стають об'єктами складних та цілеспрямованих кібератак [1]. Традиційні методи захисту виявляються недостатньо ефективними в умовах зростання обсягів даних, швидкості атак і застосування зловмисниками новітніх технологій.

Одним із перспективних напрямів посилення кіберзахисту є впровадження штучного інтелекту (ШІ), зокрема методів машинного та глибокого навчання [2]. Такі технології дозволяють підвищити рівень адаптивності, своєчасно виявляти відхилення від нормальної поведінки систем, прогнозувати можливі загрози та реагувати на інциденти з мінімальною затримкою. Метою даної роботи є аналіз підходів до використання ШІ для виявлення та протидії кібератакам на державні інформаційні системи, а також представлення узагальненої архітектури відповідної системи захисту.

Сучасні державні інформаційні системи є критично важливими елементами цифрової інфраструктури, що потребують надійного захисту від зростаючих кіберзагроз. Одним з ефективних підходів до підвищення рівня кібербезпеки є використання технологій штучного інтелекту (ШІ), які забезпечують адаптивність, швидкість реагування та високу точність виявлення атак. ШІ дозволяє автоматизувати процеси моніторингу мережевої активності, виявляючи аномалії, що свідчать про потенційні загрози. Методи машинного навчання дозволяють побудувати моделі нормальної поведінки системи та виявляти відхилення, які можуть вказувати на спроби злому, шкідливу активність або порушення політик доступу. Окрему роль відіграє глибоке навчання (Deep Learning), яке використовується для обробки великого обсягу вхідних даних у реальному часі. Це дає змогу виявляти складні мультивекторні атаки, що важко виявити за допомогою традиційних інструментів.

Штучний інтелект також ефективно використовується для:

1. Передбачення атак (predictive analytics) на основі аналізу історичних даних.
2. Аналізу поведінки користувачів (User Behavior Analytics, UBA) для виявлення внутрішніх загроз [3].
3. Інтелектуальної реакції на інциденти, включно з автоматичним ізолюванням уражених компонентів системи.

На рис. 1 наведено типову архітектуру системи кіберзахисту державної IT-інфраструктури із застосуванням ШІ.

Рис. 1. Узагальнена архітектура системи виявлення та протидії кібератакам з використанням ШІ

Після впровадження штучного інтелекту в архітектуру системи кіберзахисту державних інформаційних систем, значно зростає їх здатність до самостійного виявлення та реагування на інциденти без участі людини. Як видно з рис. 1, ШІ-орієнтована система включає в себе кілька модулів, зокрема для аналізу поведінки, виявлення аномалій, прогнозування загроз та автоматичного реагування. Така модульна побудова забезпечує гнучкість, адаптивність та можливість масштабування залежно від розміру й складності захищуваної інфраструктури.

У таблиці 1 представлено ключові методи використання штучного інтелекту в системах кіберзахисту, серед яких найбільш поширеними є виявлення аномалій, класифікація атак та автоматичне реагування [4]. Ці підходи базуються на алгоритмах машинного навчання і забезпечують вищу точність і швидкість виявлення, порівняно з традиційними засобами. Зокрема, прогнозування загроз на основі аналізу історичних даних сприяє переходу від реактивної до проактивної безпеки.

Таблиця 1

Методи використання ШІ для кіберзахисту

Метод	**Опис**
Виявлення аномалій	Побудова моделей поведінки та виявлення нетипових дій у системі
Класифікація атак	Розпізнавання типу кібератаки на основі аналізу вхідного трафіку
Автоматичне реагування	Генерація дій у відповідь на інциденти: блокування, ізоляція, сповіщення
Аналіз логів	Машинний аналіз логів для виявлення прихованих загроз
Прогнозування загроз	Використання історичних даних для виявлення майбутніх ризиків

Проведений аналіз засвідчує, що використання методів штучного інтелекту значно підвищує ефективність систем виявлення та протидії кібератакам у державних інформаційних системах. Кожен з методів, поданих у таблиці 1, виконує специфічну функцію, що в сукупності забезпечує комплексну кібербезпеку: від початкового виявлення загроз до їх нейтралізації. Особливої уваги заслуговує здатність ШІ до обробки великих обсягів даних у режимі реального часу, що дає змогу оперативно реагувати на навіть найскладніші загрози.

Разом із тим, впровадження подібних технологій супроводжується низкою викликів, пов'язаних з точністю моделей, потребою в актуальних та якісних навчальних даних, а також ризиками етичного характеру. Для досягнення максимальної ефективності необхідне поєднання технічних рішень із законодавчими та організаційними заходами, спрямованими на підтримку національної стратегії інформаційної безпеки [5].

Використання штучного інтелекту у сфері кіберзахисту державних інформаційних систем забезпечує новий рівень гнучкості, адаптивності та проактивності в боротьбі з кібератаками. Інтеграція ШІ дозволяє автоматизувати процеси виявлення загроз і реагування на них, мінімізуючи людський фактор і підвищуючи ефективність системи загалом. Подальші дослідження мають бути зосереджені на удосконаленні алгоритмів навчання, забезпеченні надійності та безпеки самих моделей, а також розробці нормативної бази, що регламентує їх використання в державному секторі.

Список літератури

1. Біленець, Д. А. (2024). Захист цифрової інфраструктури у митно-адміністративних послугах: виклики та сучасні підходи до кібербезпеки. Аналітично-порівняльне правознавство, (4), 301-306.

2. Савіцький, Л., Безносенко, С., & Горбач, Р. (2024). Концептуальні погляди на побудову системи захисту від кібератак із застосуванням методів штучного інтелекту в інформаційно-комунікаційних системах. Сучасні інформаційні технології у сфері безпеки та оборони, 49(1), 77-85.

3. G. Martín, A., Fernández-Isabel, A., Martín de Diego, I., & Beltrán, M. (2021). A survey for user behavior analysis based on machine learning techniques: current models and applications. Applied Intelligence, 51(8), 6029-6055.

4. Ali, A., Septyanto, A. W., Chaudhary, I., Al Hamadi, H., Alzoubi, H. M., & Khan, Z. F. (2022, February). Applied artificial intelligence as event horizon of cyber security. In 2022 International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-7). IEEE.

5. Ярмілко, А. В., Розломій, І. О., & Мисюра, Ю. О. (2021). Застосування хеш-методів у криптографічному аналізі потоків інформації. Вісник Хмельницького національного університету, (6), 49-54.

УДК 004.4

О. С. Ткаченко[1], А.С. Коваленко[1]
alex.tranduil@gmail.com, annasun911@gmail.com
[1]*Центральноукраїнський національний технічний університет, Кропивницький*

ОЦІНКА ЕФЕКТИВНОСТІ ВІДКРИТИХ І ВЛАСНИЦЬКИХ ІНСТРУМЕНТІВ БЕЗПЕКИ ДЛЯ LLM

Великий обсяг інформації, яку обробляють моделі машинного навчання, а також їхнє широке застосування у різних галузях призвели до суттєвого підвищення безпекових ризиків. LLM (Large Language Models) відкривають нові можливості для розв'язання складних завдань, проте вони також є об'єктом численних кіберзагроз, таких як несанкціонований доступ, маніпулювання моделями, використання у шкідливих цілях та інших [1-4].

Метою даної роботи було порівняння двох Open-source та двох Proprietary інструментів безпеки для LLM, аналізуючи їхні можливості, переваги та недоліки по виділеним критеріям.

1. Purple Llama [5] це ініціатива з відкритим кодом від Meta, орієнтована на підвищення безпеки генеративного ШІ. Вона включає в себе модуль Safety Classifier для визначення токсичного або небезпечного контенту, а також інструмент CyberSecEval для оцінки можливості генерації кіберзагроз. Purple Llama надає прозору, гнучку та відтворювану платформу для тестування LLM на відповідність вимогам безпеки.

2. Garack [6] open-source інструмент безпеки для LLM, який фокусується на виявленні prompt injection атак та небажаних інструкцій. Інтегрується з open-source моделями, надаючи API-інтерфейс для реального часу перевірки запитів та відповідей, дозволяючи виявляти потенційно шкідливі запити до LLM.

3. CalypsoAI Moderator [7-8] власницьке рішення, яке пропонує безпеку корпоративного рівня. Модератор вбудовується між користувачем і LLM, аналізує як вхідні запити, так і вихідні відповіді, класифікує ризики, блокує шкідливі запити та генерує звіти відповідності політикам. Його можна інтегрувати з будь-якою LLM-платформою, а також налаштувати політики безпеки під конкретні корпоративні потреби.

4. Lakera Guard [9] це комерційна платформа для захисту LLM, зокрема від prompt injection, витоків даних, jailbreak-атак, небезпечного або неетичного контенту. Вона підтримує детальну аналітику, real-time API, можливість інтеграції з продуктами через SDK. PRO-план забезпечує найвищий рівень захисту, у т.ч. адаптивну політику модерації в залежності від ризик-профілю компанії.

У результаті огляду були виділені критерії порівняння та сформовано таблицю 1.

Таблиця 1
Порівняльна таблиця відкритих та власницьких інструментів безпеки для LLM

Критерій	Purple Llama	Garack	CalypsoAI Moderator	Lakera Guard (PRO)
Тип ліцензії	Open-source (Apache 2.0)	Open-source (MIT)	Власницька	Власницька
Вартість	Безкоштовно	Безкоштовно	Платно (корпоративний рівень)	Платно (PRO план)
Захист від prompt injection	Обмежений	Високий	Високий	Високий
Захист від шкідливого контенту	Середній (модуль Safety)	Слабкий	Високий	Високий
Кастомізація політик	Висока	Середня	Висока (гнучкі політики під бізнес)	Висока (адаптивні політики)
Інтеграція з API	Так	Так	Так	Так
Аналітика та звітність	Базова	Відсутня	Розширена (compliance, audit, risk-звіти)	Розширена (інтерактивна аналітика, логування)
Підтримка в реальному часі	Так	Так	Так	Так
Jailbreak detection	Обмежена	Обмежена	Висока	Висока
Корпоративна придатність	Середня	Низька	Висока	Висока

Продовження таблиці 1

Критерій	Purple Llama	Garack	CalypsoAI Moderator	Lakera Guard (PRO)
Простота впровадження	Середня	Висока	Середня (може потребувати інтеграції)	Висока (готові SDK, GUI)
Налаштування і гнучкість	Висока гнучкість, потребує налаштування	Гнучкий, зручний інтерфейс	Висока, кастомізовані правила	Висока, з адаптивними конфігураціями
Підтримка і обслуговування	Спільнота, відсутня офіційна підтримка	Спільнота, немає офіційної підтримки	Професійна підтримка 24/7, SLA	Підтримка PRO-рівня, включає консультації та оновлення
Безпека і відповідність	Може потребувати додаткових налаштувань	Лише базові функції безпеки	Відповідає стандартам SOC 2, ISO/IEC, можливість сертифікації	Висока відповідність (GDPR, HIPAA, SOC 2 тощо)
Легкість у використанні	Вимагає технічних знань	Зручний інтерфейс	UI для менеджерів безпеки, може вимагати навчання	Інтуїтивно зрозумілий інтерфейс з готовими шаблонами
Оновлення	Необхідність ручного оновлення	Залежить від спільноти	Автоматичні оновлення через хмару	Автоматичні оновлення, включено у підписку PRO

Висновки Використання Purple Llama є оптимальним рішенням для створення безпечних мовних моделей завдяки його потужній відкритій основі, яка дозволяє розвивати й налаштовувати моделі відповідно до конкретних дослідницьких чи освітніх завдань. Garack вирізняється спеціалізованими можливостями захисту від атак типу prompt injection, тому він особливо корисний для інтерактивних систем та чат-ботів, де необхідно запобігати маніпулюванню запитами. Водночас CalypsoAI Moderator та Lakera Guard демонструють високу ефективність, стабільність та гнучкість, що робить їх ідеальними для комерційних та корпоративних середовищ, де безпека й відповідність вимогам регулювання є пріоритетом. Проте їхні переваги супроводжуються необхідністю суттєвих фінансових інвестицій, що обмежує їх застосування в малих проєктах або освітніх середовищах.

Таким чином, вибір відповідного інструменту залежить від балансу між бюджетом, специфічними потребами проєкту та бажаним рівнем безпеки й гнучкості.

Список літератури

1. Awesome LLM Security. URL: https://github.com/corca-ai/awesome-llm-security (дата звернення: 07.04.2025).

2. Top 12 LLM Security Tools: Paid & Free (Overwiiew). URL: https://www.lakera.ai/blog/llm-security-tools (дата звернення: 07.04.2025).

3. 10 LLM Security Tools to Know in 2025. URL: https://www.pynt.io/learning-hub/llm-security/10-llm-security-tools-to-know (дата звернення: 07.04.2025).

4. Best LLM Security Tools & Open-Source Frameworks in 2025. URL: https://www.deepchecks.com/top-llm-security-tools-frameworks/ (дата звернення: 07.04.2025).

5. Purple Llama. URL: https://github.com/meta-llama/PurpleLlama (дата звернення: 07.04.2025).

6. Garak, LLM vulnerability scanner. URL: https://github.com/NVIDIA/garak (дата звернення: 07.04.2025).

7. Secure AI at Inference. URL: https://calypsoai.com (дата звернення: 07.04.2025).

8. Why CalypsoAI. URL: https://calypsoai.com/why-calypsoai/ (дата звернення: 07.04.2025).

9. Real-Time Security for Your AI Agents. URL: https://www.lakera.ai/lakera-guard (дата звернення: 07.04.2025).

УДК 378:004

Є. В. Червоний[1], Р.О. Ткачук[1]
chervony.egor@gmail.com, tkachukroman64@gmail.com
[1]Центральноукраїнський національний технічний університет, Кропивницький

РОЛЬ DEVSECOPS РОЗРОБНИКА У ВИЯВЛЕННІ ТА ЗАПОБІГАННІ ВРАЗЛИВОСТЯМ НА РАННІХ ЕТАПАХ SSDLC

У сучасній розробці програмного забезпечення безпека повинна бути невід'ємною частиною кожного етапу життєвого циклу [1]. Концепція SSDLC (Secure Software Development Life Cycle) передбачає інтеграцію заходів безпеки на всіх фазах розробки від вимог і дизайну до вровадження та підтримки продукту [1,2]. Такий підхід дозволяє проактивно виявляти та усувати потенційні уразливості, підвищуючи загальну безпечність додатків і знижуючи ризик інцидентів [3,4]. В рамках SSDLC набуває критичного значення культура DevSecOps це практика, що об'єднує розробку, безпеку й експлуатацію [5,6]. DevSecOps розробники акцентують спільну відповідальність за безпеку і впроваджує автоматизовані засоби захисту в CI/CD-процеси [5,6]. Мета впровадження DevSecOps розробника це виявляти та усувати проблеми безпеки на найбільш ранніх стадіях розробки, запобігаючи їх появі у стадії використання користувачами продукту, де вартість виправлення помилок значно вища.

Було проведена оцінка ефективності інтеграції DevSecOps-підходів у процес SSDLC для зменшення витрат на усунення помилок безпеки у пізніх стадіях розробки продукту, зокрема розглянуто SSDLC і місце DevSecOps у безпечній розробці, вразливості на ранніх стадіях розробки, практики DevSecOps для раннього виявлення вразливостей, проаналізовані витрати на виправлення вразливостей в різні етапи розробки у результаті дана ефективність проектів з залученням DevSecOps розробників проти традиційних підходів з виділенням рекомендацій.

SSDLC це процес, який використовується індустрією програмного забезпечення для проектування, розробки і тестування високоякісного продукту. На практиці це означає врахування вимог безпеки на всіх етапах життєвого циклу. Такий процес часто називається принципом "shift left"[7] тобто зміщення заходів безпеки "вліво" на часовій шкалі SDLC, ближче до початку проектування проекту. В результаті помилки виявляються тоді, коли їх усунути найлегше і найдешевше, що дозволяє уникнути дорогого перероблення продукту перед випуском і мінімізувати затримки під час реалізації.

Важлива риса DevSecOps це спільна відповідальність всіх спеціалістів що задіяні в процес розробки, всі відповідають за безпечність коду нарівні з його функціональністю. Завдання DevSecOps розробника налаштувати процеси та інструменти таким чином, щоб безпека стала автоматично перевірятися на кожному етапі від запуску статичного аналізу коду до моніторингу на стадії використання користувачами продукту. Такий підхід забезпечує швидку розробку без жертвування безпекою, підвищує швидкість релізів продукту, водночас мінімізуючи ризики за рахунок усунення вразливостей ще на ранніх етапах розробки продукту. DevSecOps проводить інтеграцію принципів SSDLC у Agile-процеси, дозволяючи командам випускати більш якісний та захищений продукт.

Проведені дослідження дозволили сформувати список типових вразливостей на ранніх стадіях розробки продукту таких як: Неправильна обробка вводу даних; Неправильна конфігурація безпеки; Використання вразливостей компонентів; Недостатній захист автентифікації та сесій. Спільним для наведених проблем є те, що вони закладаються на ранніх етапах при написанні коду або конфігуруванні системи. DevSecOps-розробник, усвідомлюючи типові "підводні камені" ранніх стадій, впроваджує інструменти та перевірки, щоб ловити такі баги ще до того, як код злито в основну github гілку проекту.

Були визначені найпоширеніші практики DevSecOps для раннього виявлення вразливостей. Одне з ключових завдань DevSecOps розробника забезпечити автоматизоване виявлення потенційних уразливостей якнайшвидше під час розробки. Для цього в конвеєр CI/CD інтегрується цілий набір інструментів і методів: SAST (Static Application Security Testing); DAST (Dynamic Application Security Testing); SCA (Software Composition Analysis); аналіз інфраструктурного коду (IaC scanning).

Були проведено аналіз витрат на виправлення вразливостей на ранніх та пізніх етапах розробки опираючись на статистичні звіти NIST(National Institute of Standards and Technology) який показав що усунення помилок безпеки в стадії використання користувачами продукту може бути в 30 разів дорожчим, ніж виправлення тієї ж проблеми на етапі розробки. Ці витрати включають не лише прямі трудозатрати розробників, а й побічні втрати такі як перероблення архітектури, регресійне тестування, затримки, а також потенційні збитки від інцидентів безпеки.

У результаті було встановлено що запровадження DevSecOps-практик помітно впливає на якісні та кількісні показники проекту по показникам: Якість і кількість уразливостей; Витрати і ROI; Швидкість розробки і гнучкість; Вигоди для команди і процесів.

Проект з впровадженим DevSecOps підходом виграє у довгостроковій перспективі - підвищується якість продукту (менше вразливостей = вищий рівень довіри клієнтів), заощаджуються гроші (менше виправлень

постфактум і інцидентів), прискорюється вихід нових версій (конкурентна перевага на ринку), а команда працює більш злагоджено. Звісно, це вимагає початкових зусиль налаштування інструментарію, навчання людей, перебудови процесів. Проте наведені показники демонструють, що такі інвестиції виправдовуються.

Були виділені рекомендації з оптимального впровадження DevSecOps на прикладі веб-додатків в наступних напрямках: Підтримка з боку керівництва і культура безпеки; Навчання розробників безпечному кодингу; Інтеграція безпеки у кожен етап SSDLC ("shift left"); Впровадження і налаштування інструментів DevSecOps; Автоматизація та CI/CD; Регулярні аудити і покращення; Орієнтація на ризики і бізнес-цілі.

Як висновок DevSecOps-розробник відіграє ключову роль у побудові безпеки в процесі створення програмного забезпечення. Інтегруючи заходи безпеки на ранніх етапах SSDLC, він допомагає команді виявляти вразливості ще до того, як вони стануть дорогою проблемою. Аналіз показує, що проактивний підхід (shift left) значно знижує кількість дефектів, що доходять до продакшену, і кратно скорочує фінансові витрати на виправлення та усунення багу на етапі розробки або тестування коштує на порядок менше, ніж у вже випущеному продукті. Інтеграція DevSecOps практик у розробку веб-додатків підвищує якість продукту, прискорює вихід нових версій та зміцнює кіберстійкість. Зіставлення проектів з DevSecOps і без нього демонструє відчутні переваги першого, а саме: менше інцидентів, нижчі витрати, швидший реліз та краща злагодженість команди. Як недолік треба заначити що впровадження DevSecOps це еволюційний процес, що потребує часу, навчання і зміни мислення всієї команди.

Таким чином, DevSecOps у рамках SSDLC це інвестиція, яка окупається як фінансово, так і через підвищення довіри користувачів та захист репутації. Компанії, що роблять безпеку невід'ємною частиною життєвого циклу розробки, у підсумку виграють на конкурентному полі, адже можуть швидко доставляти цінність клієнтам і водночас надійно захищати їхні дані.

Список літератури

1. Secure Software Development Life Cycle (Secure SDLC). URL: https://snyk.io/articles/secure-sdlc (дата звернення: 07.04.2025).

2. Akbar M. A., Khan A. A., Mahmood S., Hyrynsalmi S. Management of DevSecOps Process: An Empirical Investigation. Software: Practice and Experience. 2025. Vol. 55. P. 1–22. DOI: 10.1002/spe.3419.

3. What is software development lifecycle security?. Red Hat, Inc. URL: https://www.redhat.com/en/topics/security/software-development-lifecycle-security (дата звернення: 07.04.2025).

4. Myrbakken H., Colomo-Palacios R. DevSecOps: Integrating Security into IT Development and Operations. Lecture Notes in Computer Science. 2017. Vol. 10027. P. 17–20. DOI: 10.1007/978-3-319-57633-6_3.

5. Deploy a comprehensive DevSecOps solution. Red Hat, Inc. URL: https://www.redhat.com/en/resources/deploy-comprehensive-devsecops-solution-overview (дата звернення: 07.04.2025).

6. Rajapakse R.N., Andrew C., Fidge C. DevSecOps: Integrating Security into the DevOps Lifecycle. International Conference on Cyber Security and Protection of Digital Services. 2023. – P. 1-10. DOI: 10.1109/CyberSecurity58641.2023.10316125.

7. Shift-left vs. shift-right testing. Red Hat, Inc. URL: https://www.redhat.com/en/topics/devops/shift-left-vs-shift-right (дата звернення: 07.04.2025).

УДК 004.056.5

В.В. Захаров[1], В.М. Чешун[1], О.В. Чешун[1], Д.А. Олексюк[2]
basistavitalij@gmail.com, cheshunvn@khmnu.edu.ua, kksmkhnu@gmail.com, oleksuk.dima@gmail.com
[1]Хмельницький національний університет, Хмельницький
[2]Хмельницький фаховий економіко-технологічний коледж, Хмельницький

ТЕХНОЛОГІЇ МЕРЕЖЕВИХ ПРИМАНОК І ЇХ ПОТЕНЦІАЛ В ЗАХИСТІ КОРПОРАТИВНОЇ ІНФОРМАЦІЇ

Сучасні кіберзагрози стають дедалі складнішими і підходять до атак з використанням новітніх технологій, тому традиційних методів захисту, як-от міжмережеві екрани та антивіруси, уже недостатньо. У цьому контексті інноваційні технології, такі як мережеві приманки, забезпечують додатковий рівень захисту, дозволяючи перехоплювати та аналізувати атаки на ранніх стадіях.

Технології приманок у мережі (Deception technology) – це спеціальний метод відволікання кіберзлочинців від реальних ресурсів компанії, спрямовуючи їх приманкою у заздалегідь створені пастки. Ці приманки імітують справжні сервери, програми та дані, створюючи у зловмисників ілюзію доступу до критично важливих активів, хоча насправді вони взаємодіють із фальшивими об'єктами [1].

У боротьбі між хакером і кіберзахисником загальноприйнято вважати, що правопорушник має перевагу: кіберзахисники мають переконатися, що все належним чином обслуговується та запобігати вторгненням у кожній точці, тоді як хакерам просто потрібно скористатися однією вразливістю для порушення захисту . У той же час зловмисники завжди можуть отримати інформацію про цільову систему або мережу за допомогою різноманітних тактик розвідки та виявлення, тоді як захисникам зазвичай не вистачає інформації про своїх противників. Очікується, що такі асиметричні недоліки для кіберзахисту будуть перебалансовані шляхом використання оборонного обману, який, як очікується, матиме кардинальний вплив на те, як протистояти загрозам [2].

Стратегія захисту на основі периметра з використанням звичайних заходів безпеки, таких як брандмауери, засоби контролю автентифікації та системи запобігання вторгненням, виявилася слабкою проти проникнення. Навіть із стратегією поглибленого захисту Міністерства внутрішньої безпеки США (2016) [3], де кілька рівнів звичайних засобів контролю безпеки розміщено по всій цільовій мережі, кіберзахисникам усе ще важко запобігати та виявляти складні атаки, такі як APT. Такі цілеспрямовані атаки зазвичай використовують уразливості нульового дня, щоб закріпитися на цільовій мережі та залишити дуже мало слідів своєї зловмисної діяльності для виявлення. Крім того, звичайні рішення для виявлення аномалій, такі як системи виявлення вторгнень і сканери зловмисного програмного забезпечення на основі поведінки, як правило, викликають переважну кількість хибно-позитивних сповіщень, що завдає шкоди кіберзахисникам і знижує їхню ефективність у виявленні справжніх атак і реагуванні на них. Захисний обман, який характеризується здатністю виявляти вразливості нульового дня та низьким рівнем помилкових тривог завдяки чіткій межі між законною діяльністю користувача та зловмисною взаємодією, може діяти як додатковий рівень захисту для пом'якшення проблем[2].

Замість того, щоб зосереджуватися на діях зловмисників, технологія мережевих приманок працює на їхньому сприйнятті, затьмарюючи поверхню атаки. Мета полягає в тому, щоб приховати цінну корпоративну інформацію від зловмисників і заплутати або ввести їх в оману, тим самим збільшуючи ризик їх виявлення, змушуючи їх неправильно спрямовувати або витрачати ресурси, затримуючи ефект атак і передчасно викриваючи торговельні дії противника . Іншими словами, захисний обман допомагає встановити активну позицію кіберзахисту, де ключовими елементами є передбачення атак до того, як вони відбудуться, збільшення витрат супротивника та збір нових даних про загрози для запобігання подібним атакам. Технологія мережевих приманок має на меті ввести зловмисників в оману, змушуючи їх вважати, що вони успішно проникли в систему. Наприклад, хакери можуть думати, що виконують атаку з розширенням привілеїв, тоді як насправді вони взаємодіють лише з фальшивими інструментами, не отримуючи реального доступу до привілейованих прав чи впливу на інфраструктуру.

Приманка у мережах імітує законні сервери, програми та дані, щоб зловмисник обманом був змушений повірити, що він проник і отримав доступ до найважливіших активів підприємства, хоча насправді це не так. Стратегія використовується для мінімізації збитків і захисту справжніх активів організації. Одним із ризиків цієї технології є те, що сучасні атаки стають дедалі масштабнішими та складнішими і сервер обману разом із фіктивними активами може виявитися недостатньо потужним, щоб впоратися з ними. До того ж досвідчені кіберзлочинці можуть швидко зрозуміти, що їх обманюють, адже приманки та фальшиві ресурси можуть видатися їм підозрілими. У такому випадку вони можуть припинити атаку та повернутися з новими, ще витонченішими методами.

Для ефективності технологія мережевих приманок повинна залишатися непомітною не лише для кіберзлочинців, але й для співробітників компанії, підрядників і клієнтів. Її механізм полягає у створенні

імітованих цифрових активів, схожих на реальні ресурси, що є в інфраструктурі організації. Коли хакери намагаються взаємодіяти з цими фальшивими елементами, вони фактично потрапляють у пастку, не завдаючи шкоди критично важливим системам.

Зазвичай технологія мережевих приманок не є основною складовою кібербезпеки, однак її використовують для реагування на можливі загрози. Її мета – запобігти несанкціонованому доступу, перенаправивши зловмисників на фіктивні дані. Це дозволяє забезпечити безпеку справжніх ресурсів підприємства. Крім того, технологія мережевих приманок є ефективним інструментом для вивчення поведінки кіберзлочинців. Аналіз їх дій під час проникнення та взаємодії з фальшивими даними дозволяє спеціалістам з кібербезпеки глибше розуміти методи атак. Деякі організації навіть розгортають спеціалізовані сервери обману, які фіксують усі дії зловмисників – від моменту вторгнення до їхніх маніпуляцій з приманкою. Такий підхід допомагає відстежувати вектори атак і надавати цінну інформацію для вдосконалення системи безпеки та попередження подібних загроз.

Іншим важливим елементом технології мережевих приманок є система сповіщень, яка дозволяє фіксувати активність зловмисника в реальному часі. Як тільки сервер отримує відповідне сповіщення, він починає записувати всі дії хакера в зоні, на яку спрямована атака. Це забезпечує можливість глибокого аналізу методів і тактик, що використовуються кіберзлочинцями, надаючи цінну інформацію для вдосконалення засобів захисту.

Однією з ключових переваг технологій мережевих приманок є їх здатність виявляти активи, які найбільше приваблюють зловмисників. Наприклад, можна припустити, що бази даних із чутливою інформацією, такою як платіжні реквізити, особисті дані чи номери соціального страхування, є основними цілями хакерів. Однак за допомогою технологій обману можна точно підтвердити, які саме активи найчастіше стають об'єктами атак. Більше того, технологія мережевих приманок дозволяє визначати конкретні типи даних, які цікавлять зловмисників. IT-команди можуть створювати симульовані середовища з різними типами підробленої інформації, наприклад бази даних, що містять вигадані номери соціального страхування, імена, адреси чи навіть облікові дані керівників компанії. Спостерігаючи за тим, до яких саме активів отримують доступ хакери, можна зробити висновки щодо їхніх цілей та пріоритетів.

Технологія мережевих приманок має низку переваг і залишається важливим компонентом сучасних стратегій кібербезпеки. Однією з ключових переваг є здатність значно скоротити час перебування зловмисника в мережі. Це досягається завдяки використанню активів-приманок, які виглядають достатньо переконливо, щоб хакери прийняли їх за реальні. Проте, коли атака починає поширюватися, IT-фахівці можуть втрутитися, обмежуючи доступ зловмисників до мережі.

У певний момент хакери можуть зрозуміти, що мають справу із симульованими активами, і усвідомити, що всі справжні ресурси організації залишаються поза їхнім доступом. Це може змусити їх припинити атаку та залишити мережу, визнавши операцію невдалою. Завдяки цьому якісна реалізація технологій приманок сприяє не лише збору цінних даних про атаки, а й зменшенню шкоди, яку може завдати зловмисник. Хоча зломи завжди є небажаними, аналіз точок входу та поведінки кіберзлочинців під час атаки забезпечує цінні дані для аналітиків. Зібрана інформація допомагає не лише зміцнити мережеву інфраструктуру, а й розробити більш ефективні стратегії захисту від майбутніх атак. Чим переконливіше налаштовані приманки, включаючи сервери, програми та підроблені дані, тим довше триває симульована атака. Це збільшує обсяг отриманих даних, необхідних для вдосконалення захисту.

Технологія мережевих приманок дозволяє IT-командам детально досліджувати поведінку зловмисників під час атак на приманні активи. Завдяки ресурсам, які зосереджуються на аналізі цих атак, такі інциденти розглядаються як важливі місії. Це дає змогу швидко реагувати на несанкціонований доступ або аномальну активність у мережі. Технологія обману скорочує середній час виявлення та нейтралізації загроз. Одним із ключових викликів кібербезпеки є велика кількість сповіщень, які можуть перевантажити IT-команду. Використовуючи технологію мережевих приманок, команда отримує лише цільові сповіщення, коли зловмисники перетинають периметр захисту та взаємодіють із приманками.

Масштабування технології приманок є відносно простим та економічно вигідним процесом. Один сервер-приманку можна багаторазово використовувати, а створення фальшивих даних, таких як вигадані облікові записи чи паролі, не потребує значних ресурсів. Крім того, автоматизовані інструменти, що використовуються в інших елементах кіберзахисту, можуть бути інтегровані й у технологію обману.

Список літератури

1. Жилін А., Шевчук О. Архітектура та класифікація Deception Technology. Information Technology and Security. 2021. Vol. 9, Iss. 2 (17). P. 165–175.

2. Li Zhang, Vrizlynn L.L. Thing. Three decades of deception techniques in active cyber defense - Retrospect and outlook. Computers & Security. 2021. Volume 106. P. 1-19.

3. Abdelghani Tschroub. Implementation of Defense in Depth Strategy to Secure Industrial Control System in Critical Infrastructures. American Journal of Artificial Intelligence. 2020. № 3. P. 17-22.

УДК 004.056.5

В.А. Басистий[1], В.М. Чешун[1], Д.В. Чешун[2], А.Р. Школьняк[1]
basistavitalij@gmail.com, cheshunvn@khmnu.edu.ua, dmytro.cheshun@gmail.com, androidartem2014@gmail.com
Хмельницький національний університет, Хмельницький
[2]Хмельницький фаховий економіко-технологічний коледж, Хмельницький

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МІКРОКОМП'ЮТЕРНОЇ СИСТЕМИ ЗАХИСТУ ІОТ

Системи IoT, поряд з широким розповсюдженням, стрімким розвитком і великими перспективами на майбутнє, мають значний недолік – більшість пристроїв не здатні виконувати складні завдання інформаційної безпеки, такі як, наприклад, моніторинг мережевого трафіку[1]. Це пояснюється їхніми обмеженими ресурсами: низькою обчислювальною потужністю, невеликим обсягом пам'яті та обмеженою енергетичною ємністю. Через ці фактори встановлення засобів для постійного моніторингу та аналізу даних безпосередньо на IoT-пристроях стає неможливим. Численні вразливості роблять мережі IoT популярною мішенню для кіберзлочинців[2,3].

Внаслідок таких обмежень завдання з моніторингу часто передаються на централізовані сервери або хмарні платформи, які мають достатньо ресурсів для обробки великих обсягів даних. Однак і цей підхід має свої недоліки. По-перше, безперервне передавання даних від великої кількості пристроїв до сервера може спричинити перевантаження мережі, особливо якщо обсяг переданої інформації значний. По-друге, централізована система створює ризик повної зупинки у разі виходу з ладу сервера або його компрометації, що робить всю мережу вразливою до атак.

Крім того, обробка даних на централізованих серверах може бути недостатньо швидкою для IoT-систем, які потребують миттєвої реакції на загрози. Затримки, пов'язані з передаванням інформації до сервера та отриманням зворотного зв'язку, підвищують ймовірність успішної атаки та негативно впливають на загальну безпеку[4].

Таким чином, обмеження архітектури IoT вимагають нових підходів до забезпечення безпеки. Нобхідно розробляти рішення, які дозволять проводити локальний аналіз та захист даних на рівні самих пристроїв, враховуючи їхні обмежені ресурси.

В [5] авторами запропоноване рішення для створення комплексу моніторингу і аналізу мережевого трафіку IoT на одноплатних мікрокомп'ютерах.

Базовим елементом комплексу є мікрокомп'ютер Raspberry Pi (можлива реалізація на інших одноплатних мікрокомп'ютерах), на якому розгортається програма моніторингу. Для розробки програмної складової комплексу було обрано мову програмування Python з бібліотеками Psutil, Scapy, Pandas, що дає можливість отримувати детальну інформацію про активні процеси та підключені пристрої. Сховищем інформації виступає сервер, який дозволяє користувачу збирати окремі пакети за стандартним протоколом від певного пристрою або від всіх одночасно. Аналіз проводиться на комп'ютері користувача без використання ресурсу основного пристрою.

Після визначення концепції організації і функціонування мікрокомп'ютерної системи захисту IoT постала задача розробки програмного забезпечення зазначеної системи.

Для моніторингу і аналізу трафіку, була створена утиліта мовою програмування Python з використанням бібліотек Psutil, Scapy, Pandas [4] на базі ядра Linux, що надає можливість запускати певну кількість пристроїв для моніторингу, яка буде потрібна користувачеві, не обмежуючись пристроями, які можуть не підтримуватись програмно або апаратно операційною системою Windows або MacOS.

Структура бази даних комплексу моніторингу і аналізу мережевого трафіку IoT представлена на рис. 1.

Розглянемо призначення основних таблиць бази даних, що наведені на рис.1

Таблиця «IoT-пристрої» ідентифікує наявні пристрої IoT і містить такі параметри: Id пристрою, назва і опис пристрою. Ця таблиця надає основну інформацію про трафік даних, які можуть бути використані для аналізу і вдосконалення продуктивності мережі.

Таблиця «Активні процеси» містить інформацію про Id процесу, назву процесу, час створення і його опис. таблиця містить інформацію про завантаження даних, вивантаження даних, швидкість завантаження і вивантаження, час виміру, назву даних, тип даних і приклад запису даних. Ця таблиця надає основну інформацію про трафік даних, які можуть бути використані для аналізу і вдосконалення продуктивності мережі.

Таблиця «Активні програми» є проміжною між процесами та пристроями IoT і містить дані для ідентифікації програмного продукту, що має вразливості або несе загрози.

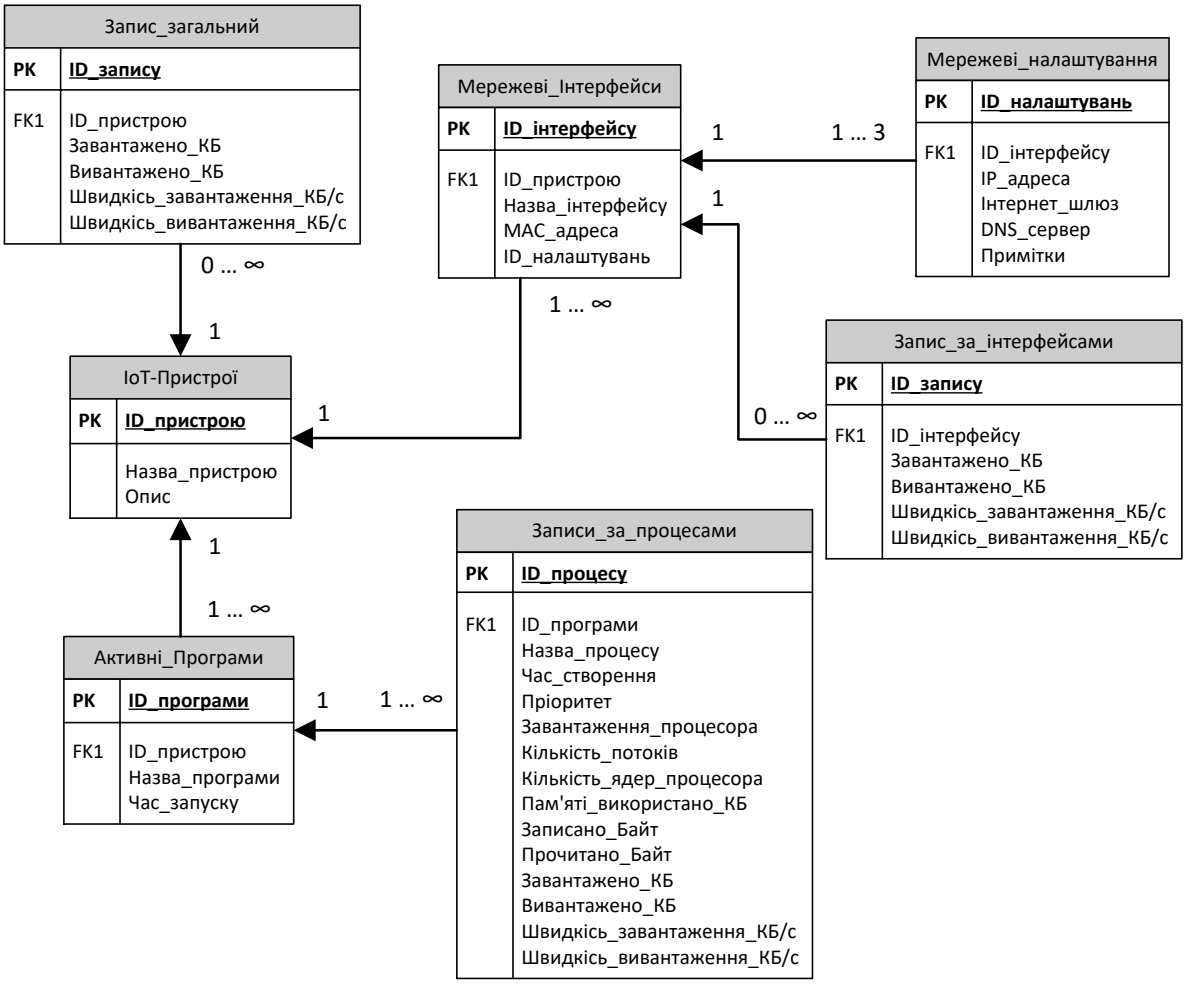


Рис. 1. База даних комплексу

Таблиці «Мережеві інтерфейси» і «Запис за інтерфейсами» містять інформацію мережеві інтерфейси і про параметри трафіку мережевих інтерфейсів, за яким можна визначити завантаженість інтерфейсу: Mac-адресу, IP-адресу, Інтернет шлюз, DNS сервер і опис. Ці параметри можуть оновлюватись і добавлятись в залежності від даних, які приймаються пристроями. Основні параметри таблиці «Мережеві інтерфейси», такі як: Id процесу, назва процесу, час створення є початковою необхідністю, за якою визначають процес. За допомогою даних цих таблиць можна визначити кожен процес, який створюється і здійснити класифікацію процесу. Таблиця «Запис за інтерфейсами» надає основну інформацію про інтерфейс, за якою можна класифікувати і виявляти інциденти, які можуть виникнути на цьому рівні.

Запропоновані утиліта моніторингу мережевого трафіку і база даних орієнтовані на використанні в роботі комплексу моніторингу і аналізу мережевого трафіку IoT на одноплатних мікрокомп'ютерах. Такий комплекс може використовуватись в малому бізнесі, коли немає можливості оплатити потужний пакет захисту з обладнанням, або виступати його тимчасовою альтернативою.

Список літератури

1. Засоби моніторингу мережі в IoT інфраструктурі з гібридною архітектурою / Каплунов А. В., та ін. Телекомунікаційні та інформаційні технології. 2023. № 2(79). С.22-32.

2. Власенко М., Хлапонін Ю. Інтернет речей (IoT) у світовій практиці: огляд та аналіз. Pidvodni Tehnologii, 2024. №13. С.21-27.

3. Проблеми та загрози безпеці IoT пристроїв / Opirskyy, I. та ін. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2021. №3(11). С. 31-42.

4. IoT Monitor Traffic: Unveiling a Smarter Approach to Monitoring Traffic. Temok.com. URL: https://www.temok.com/blog/iot-monitor-traffic/. (date of access:30.03.2025).

5. Басистий В.А., Чешун О.В., Чешун В.М. Комлекс моніторингу і аналізу мережевого трафіку IOT на одноплатних мікрокомп'ютерах. Тези доповідей XXVII Всеукраїнської НПК «Могилянські читання – 2024», Технічні науки. С.103-108.

УДК 004(056.53+413.4)::001.51

В.В. Мохор[1], О.О. Бакалинський[1], Я.Ю. Дорогий[2], В.В. Цуркан[1,3]
v.mokhor@gmail.com, baov@meta.ua, yaroslav.dorohyi@donntu.edu.ua, v.v.tsurkan@gmail.com
[1]Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова Національної академії наук України, Київ
[2]Донецький національний технічний університет, Дрогобич
[3]Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ

СПОСОБИ ОБИРАННЯ МЕТОДУ ОЦІНЮВАННЯ РИЗИКІВ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Відповідно до [1], організаціям незалежно від типу, розміру, сфери діяльності необхідно визначати та застосовувати процес поводження з ризиками. Насамперед це стосується розробляння і впроваджування інтегрованої системи управління інформаційною безпекою, кібербезпекою, приватністю, використанням штучного інтелекту [2, 3]. Визначенню її складників передує задання критеріїв прийнятності та оцінювання ризиків і, як наслідок, обирання відповідного методу на етапі встановлювання контексту [1, 4, 5]. Цим ураховуються як потреби, очікування, обмеження з боку зацікавлених сторін, так і особливості розробляння і впроваджування інтегрованої системи управління безпекою діяльності організації. Тож обирання методу оцінювання ризиків інформаційної безпеки є актуальним завданням.

Розробляння інтегрованої системи управління безпекою діяльності організації обумовлюється внутрішнім і зовнішнім контекстами. Це спонукає до врахування насамперед меж її впроваджування (частина організації, організація загалом); наявності інформації про вразливості інформаційних активів, заходів/засобів, загрози, наслідки реалізування загроз і мети оцінювання ризиків. Тому обирання відповідного методу може здійснюватися двома способами [5]. Першим з них визначаються і ураховується вплив відповідних характеристик з огляду на контекст діяльності організації. Тоді як другим – завдання оцінювання ризиків інформаційної безпеки. Приклад обирання методу аналізування впливу на приватність (англ. Privacy Impact Analysis, PIA) другим способом представлено табл. 1 [5] з урахуванням настанов [1, 4]. Так, він може застосовуватися для ідентифікування, визначання оцінок вірогідності, рівня ризику. До того ж завжди застосовується у межах виконання завдань визначання оцінок наслідків реалізування загрози та зіставляння ризиків інформаційної безпеки [4].

Таблиця 1

Приклад використання способу обирання методу оцінювання ризиків інформаційної безпеки відповідно до завдань

Метод оцінювання ризиків інформаційної безпеки	Оцінювання ризиків інформаційної безпеки				
	Ідентифікування ризиків інформаційної безпеки	Аналізування ризиків інформаційної безпеки			Зіставляння ризиків інформаційної безпеки
		Наслідки	Вірогідність	Рівень ризику	
Аналізування впливу на приватність	Застосовується	Завжди застосовується	Застосовується	Застосовується	Завжди застосовується

Отже, запорукою результативності розробляння і впроваджування інтегрованої системи управління безпекою є урахування внутрішнього і зовнішнього контекстів діяльності організацій. Це досягається встановлюванням критеріїв прийнятності, оцінювання ризиків інформаційної безпеки та, як наслідок, обиранням відповідного методу одним з або комбінуванням способів на основі характеристик і завдань.

Список літератури

1. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection. Information security management systems. Requirements. [Valid from 2022-10-25]. URL: https://www.iso.org/standard/27001 (accessed on: 08.04.2025).

2. The Integrated Use of Management System Standards (IUMSS). [Valid from 2018-11]. URL: https://www.iso.org/publication/PUB100435.html (accessed on: 08.04.2025).

3. Мохор В. В., Бакалинський О. О., Дорогий Я. Ю., Цуркан В. В. Симбіоз систем захисту інформації об'єктів критичної інфраструктури сфери енергетики. *Кібербезпека енергетики* : матеріали науково-практичної конференції (Київ, 31 травня 2023 р.). Київ, 2023. С. 107–108. DOI: https://doi.org/10.5281/ zenodo.14601428

4. ISO/IEC 27005:2022. Information security, cybersecurity and privacy protection. Guidance on managing information security risks. [Valid from 2022-10-25]. URL: https://www.iso.org/standard/80585.html (accessed on: 08.04.2025).

5. IEC 31010:2019. Risk management. Risk assessment techniques. [Valid from 2019-06-17; revised 2024-09-03]. URL: https://www.iso.org/standard/72140.html (accessed on: 08.04.2025).

УДК 004.9:681.3

В.В. Ситенкова, *ст. 2-го курсу*
kin77444@gmail.com
Науковий керівник: Л.В. Константинова, *викладач*
Центральноукраїнський національний технічний університет, Кропивницький

ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ МЕТОДІВ ШИФРУВАННЯ ДАНИХ В УМОВАХ РОЗВИТКУ КВАНТОВИХ ОБЧИСЛЕНЬ

У сучасному світі безпека даних є критично важливою складовою функціонування держав, компаній та особистого життя користувачів. Сьогоднішні методи шифрування забезпечують високий рівень захисту, однак розвиток квантових обчислень ставить під загрозу їхню надійність. Квантові комп'ютери мають потенціал кардинально змінити уявлення про криптографічну безпеку, оскільки здатні ефективно вирішувати завдання, які класичні комп'ютери розв'язують мільйони років. Тому дослідження застосування методів шифрування даних в умовах розвитку квантових обчислень буде актуальною задачою.

Особливу загрозу становлять квантові алгоритми, зокрема алгоритм Шора, який дозволяє швидко факторизувати великі числа, що лежать в основі сучасних асиметричних криптографічних систем. Це може призвести до того, що методи, які нині вважаються безпечними, стануть непридатними для захисту конфіденційної інформації.

Важливо зауважити, що на даний момент не існує достатньо потужного квантового комп'ютера, але його появу прогнозують у найближчі десятиліття [1]. Це дає обмежений час для підготовки до нової ери обчислень і розробки стійких до квантових атак алгоритмів.

Сучасні криптографічні методи поділяються на декілька основних категорій: асиметричне шифрування, симетричне шифрування, гібридні методи та хеш-функції. Кожен з цих підходів має свої унікальні характеристики, принципи роботи, а також рівень стійкості до різних типів атак. Розуміння їхніх особливостей є ключовим для формування ефективної стратегії захисту інформації в умовах динамічного розвитку обчислювальних технологій, особливо коли йдеться про потенційні загрози з боку квантових комп'ютерів.

Асиметричне шифрування базується на використанні пари ключів: відкритого та закритого. Найпоширенішими алгоритмами є RSA та ECC. Згідно з [2], для зламу RSA-ключа довжиною 2048 бітів звичайним комп'ютером потрібно близько 19,8 квадриліонів років (табл.1). Водночас квантовий комп'ютер здатен здійснити цю операцію всього за 8 годин [3]. Аналогічно, алгоритми на базі еліптичних кривих (ECC) вважаються практично незламними для класичних комп'ютерів [4], однак квантові здатні впоратись із цим завданням за кілька хвилин [4]. Це пов'язано з ефективністю алгоритму Шора, що дозволяє швидко виконувати факторизацію та дискретне логарифмування.

Симетричне шифрування, зокрема AES (Advanced Encryption Standard), використовує один спільний ключ для шифрування і дешифрування. Алгоритм AES-128 вважається стійким до класичних атак: для його зламу потрібні приблизно 2,158 трильйонів років [5] (табл.1). Проте при використанні квантових алгоритмів, таких як алгоритм Гровера, цей час може скоротитись до 600 років [6]. Хоча симетричні методи є менш вразливими до квантових атак, вони також потребують удосконалення або збільшення довжини ключів.

Гібридні методи поєднують асиметричне та симетричне шифрування, використовуючи сильні сторони обох підходів. Такі системи частіше зустрічаються в практичних реалізаціях, наприклад, у TLS-протоколах. Проте, враховуючи вразливість асиметричної частини до квантових атак, стійкість таких методів також перебуває під сумнівом у майбутньому.

Таблиця 1

Порівняння зламу звичайним та квантовим комп'ютерами.

Алгоритм	**Довжина**	**Час зламу звичайним комп'ютером**	**Час зламу квантовим комп'ютером**
RSA	2048	19,8 квадриліонів років [2]	8 годин [3]
ECC	256	Практично неможливо [4]	Кілька хвилин [4]
AES	128	2,158 трильйонів років [5]	600 років [6]

Хеш-функції, зокрема SHA-2 та SHA-3, використовуються для перевірки цілісності даних та цифрових підписів. Вони також можуть бути піддані атакам з боку квантових алгоритмів, але загалом вважаються менш вразливими, ніж асиметричне шифрування.

Узагальнюючи, можна сказати, що більшість існуючих методів шифрування певною мірою вразливі до потенційних атак із боку квантових комп'ютерів. Це особливо стосується асиметричних алгоритмів, які можуть бути зламані за порівняно короткий час. Тому вже сьогодні важливо звернути увагу на нові підходи до захисту інформації, адаптовані до умов майбутніх квантових обчислень. до квантових атак. Це підкреслює необхідність

розвитку та впровадження постквантових криптографічних рішень, що забезпечать безпеку в умовах розвитку нових технологій, зокрема квантових обчислень.

У відповідь на загрозу, яку становлять квантові комп'ютери, розпочато розробку нових криптографічних алгоритмів, що мають забезпечити захист інформації навіть за наявності потужних квантових обчислень. Ці алгоритми отримали назву постквантових. Перехід до постквантової криптографії має розпочатися вже зараз, щоб уникнути ризиків у майбутньому. Серед рекомендованих напрямків - поступовий перехід до нових алгоритмів, проведення тестування їх ефективності та інтеграція у важливі цифрові інфраструктури [1].

Сфера інформаційної безпеки має безпосередній вплив на функціонування критичної інфраструктури: банківських систем, медичних установ, державного управління та навіть побутових пристроїв. Уразливість криптографії може призвести до масштабних витоків даних, втрати коштів або блокування доступу до життєво важливих сервісів. Саме тому питання адаптації до нових загроз не є суто теоретичним, а потребує практичних рішень уже сьогодні.

З огляду на швидкий розвиток технологій, фахівці з кібербезпеки, науковці та державні структури повинні тісно співпрацювати у впровадженні надійних рішень. Постквантова криптографія - це не лише новий етап технічного прогресу, а й виклик, що потребує об'єднання зусиль на глобальному рівні. Розуміння масштабів загрози й активна підготовка до змін є основою для збереження довіри до цифрового простору в майбутньому. Це дозволить забезпечити довготривалу захищеність інформаційних систем навіть в умовах настання квантової ери.

У роботі було проаналізовано сучасні методи шифрування та їхню стійкість до квантових атак. Було з'ясовано, що асиметричні алгоритми (RSA, ECC) є найбільш вразливими, а симетричні методи (AES) мають відносно вищу стійкість, хоча також потребують посилення.

Незважаючи на те, що повноцінні квантові комп'ютери ще не створені, розвиток цієї галузі вже впливає на підходи до інформаційної безпеки. Поява постквантових алгоритмів - важливий крок до збереження конфіденційності та цілісності даних у майбутньому.

Список літератури

1. Cyber chiefs unveil new roadmap for post-quantum cryptography migration. NCSC. URL: https://www.ncsc.gov.uk/news/pqc-migration-roadmap-unveiled (date of access: 16.04.2025).

2. Carlius A. Quantom encryption - Time to crack? (3/3). LinkedIn: Log In or Sign Up. URL: https://www.linkedin.com/pulse/quantom-encryption-time-crack-33-anders-carlius (date of access: 16.04.2025).

3. Gidney C., Ekerå M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum. 2021. Vol. 5. P. 433. URL: https://doi.org/10.22331/q-2021-04-15-433 (date of access: 16.04.2025).

4. Quantum vs. regular computing time to break ECC?. Cryptography Stack Exchange. URL: https://crypto.stackexchange.com/questions/35384/quantum-vs-regular-computing-time-to-break-ecc (date of access: 16.04.2025).

5. How long would it take to brute force an AES-128 key?. Cryptography Stack Exchange. URL: https://crypto.stackexchange.com/questions/48667/how-long-would-it-take-to-brute-force-an-aes-128-key (date of access: 16.04.2025).

6. National Academies of Sciences, Engineering, and Medicine. Quantum computing: progress and prospects. National Academies Press, 2019.

УДК 004.8

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

ОГЛЯД МОЖЛИВОСТЕЙ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ГАЛУЗІ КІБЕРБЕЗПЕКИ

У добу стрімкого науково-технічного прогресу світ щодня ставить перед суспільством нові виклики. Глобальна цифровізація призводить до стрімкого зростання обсягів інформації та її оброки, хоч значно її частина все ще і залишається необробленою. Отже спостерігається постійно ростуча потреба у надійному її захисті, отже значно підвищується об'єм таких робіт, відповідно професіоналізм фахівців з кібербезпеки має рости. У цьому контексті, штучний інтелект (ШІ) має виступити інноваційним інструментом професіонала, що ефективно виявлятиме загрози при всебічній автоматизації процесів захисту систем.

ШІ вже сьогодні виступає потужним інструментом - як для фахівців, так і, на жаль, для зловмисників. Хакери активно застосовують технології ШІ для автоматизації процесів розвідки, виявлення та використання "слабких місць" у мережах. У той же час фахівці з кібербезпеки часто перевантажені великим обсягом робіт, які необхідно оперативно виконувати. У зв'язку з цим варто вже зараз звернути увагу на можливості ШІ з позиції його використання у конкретній сфері застосування - у кібербезпеці.

Сучасні умови розвитку інформаційних технологій вимагають від фахівців високого рівня взаємодії зі ШІ. Вже зараз ШІ виступає дієвим інструментом для виявлення систем вторгнення, аналізу лог-файлів, фільтрації спам-повідомлень та ідентифікації потенційно небезпечного трафіку. Фахівець із кібербезпеки має володіти навичками роботи з віртуальними помічниками, які здійснюватимуть автоматизовану реакцію загрози.

Через велику завантаженість можна своєчасно й не помітити загрозу що приводить до витоку інформації чи інших негативних наслідків. На допомогу прийде ШІ, який виявить незвичні дії, атаки "нульового дня" чи шкідливі види програмного забезпечення. Віртуальний помічник може обробляти "логи", фільтрувати спам чи атаки.

Щоб працювати в сфері кібербезпеки з ШІ, для початку важливо розуміти, що таке машинне навчання та як воно працює. Ці знання дозволять створювати системи, здатні до самостійного аналізу загроз і адаптації до нових умов. Оволодіння навичками роботи з алгоритмами класифікації і кластеризації сприятиме ефективному будуванню систем захисту, які реагуватимуть на загрози у режимі реального часу.

Згідно з нещодавніми дослідженнями, галузь кібербезпеки відчуває дефіцит спеціалістів, здатних ефективно використовувати можливості ШІ: виявлення загроз та розробки AI-моделей. Фахівці, які володіють знаннями у сфері ШІ, вважаються більш конкурентоспроможними на ринку праці, а працедавці вже зараз шукають таких спеціалістів.

На завершення хочу зазначити, що внаслідок тезового представлення інформації, окреслено лише основні напрямки застосування ШІ в сфері кібербезпеки. Водночас бракує прикладів його практичного використання у відповідності до сфери застосування. Також, хочу наголосити, що для майбутніх фахівців з кібербезпеки надзвичайно важливим є вивчення та аналіз актуальних кейсів, а також ознайомлення з провідними напрямками розвитку ШІ, що сприятиме їхньому професійному становленню та розумінню сучасних тенденцій розвитку науки.

Список літератури

1. Кібербезпека та штучний інтелект - Web Academy Media. Web Academy Media. URL: https://web-academy.ua/blog/junior/cybersecurity-and-artificial-intelligence (дата звернення: 15.04.2025).

2. Україна В. Переваги використання Штучного інтелекту в кібербезпеці. *Міжнародна аудиторська компанія BDO - BDO*. URL: https://www.bdo.ua/uk-ua/insights-2/information-materials/2024/corporate-cybersecurity-ai-role-in-data-protection (дата звернення: 15.04.2025).

3. Передові професії у сфері штучного інтелекту: ваш шлях до успіху в AI. *InProject - IT компанія, яка створює неймовірні digital продукти*. URL: https://inproject.org/top-10-profesij-u-sferi-shtuchnogo-intelektu/ (дата звернення: 15.04.2025).

СЕКЦІЯ 2. ПРОГРАМУВАННЯ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.75

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

ІНЖЕНЕРІЯ ТЕСТОВАНОСТІ: ІНДЕКС ТЕСТОВОЇ ПРОНИКНОСТІ ЯК ІНСТРУМЕНТ ОЦІНКИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ. Тестування програмного забезпечення є ключовим елементом інженерії програмних продуктів, спрямованим на забезпечення їхньої якості. Тестованість, як здатність системи бути перевіреної, відіграє вирішальну роль у цьому процесі, впливаючи на ефективність виявлення дефектів і стабільність кінцевого продукту [1]. Оцінка якості тестування дозволяє визначити, наскільки тести відповідають цілям розробки, що особливо важливо в умовах зростання складності сучасних програмних систем [2]. У цьому контексті актуальними стають нові теоретичні підходи, які враховують архітектурні особливості програмного забезпечення.

Перспективні напрямки включають використання машинного навчання для аналізу коду та прогнозування зон із потенційними дефектами [3], а також розробку формальних моделей для оцінки тестованості [4]. Такі методи формують основу для інноваційних рішень у інженерії програмного забезпечення, спрямованих на оптимізацію процесу верифікації.

Основна частина. Запропоновано ідею "індексу тестової проникності" (ІТП) — нового методу оцінки тестованості програмного забезпечення, що базується на аналізі взаємодії між компонентами системи та їхньою доступністю для тестування. ІТП є кількісною мірою, яка показує, наскільки легко тести можуть "проникати" через архітектуру програми, враховуючи її структурні особливості та залежності.

Концепція ІТП спирається на ідею, що тестованість залежить не лише від покриття коду чи модульності, а й від доступності внутрішніх компонентів через точки входу. У системах із високою щільністю залежностей тести часто обмежуються поверхневими шарами, залишаючи глибші компоненти неперевіреними, де можуть ховатися критичні дефекти [5]. ІТП оцінює проникність через три параметри:

Щільність залежностей: Кількість і складність зв'язків між компонентами, що ускладнюють доступ тестів до модулів.

Глибина архітектури: Відстань від точок входу (API, інтерфейси) до найвіддаленіших компонентів.

Тестові бар'єри: Ізольовані чи погано документовані компоненти, що створюють перешкоди для тестування.

Формально ІТП виражається як:

ІТП=F(ЩЗ, ГА, ТБ), де ЩЗ — щільність залежностей, ГА — глибина архітектури, ТБ — тестові бар'єри, а F — функція (наприклад, зважена сума). Значення ІТП варіюється від 0 (неможливість тестування) до 1 (максимальна проникність).

Для реалізації ІТП пропонується графовий аналіз: код представлено як граф із вершинами (модулі) та ребрами (залежності). Щільність обчислюється як відношення ребер до вершин, глибина — як максимальна довжина шляху, бар'єри — як кількість недоступних вершин. Цей підхід може бути автоматизований за допомогою інструментів статичного аналізу [6].

Розширення концепції ІТП передбачає аналіз не лише статичних характеристик коду, а й динамічних аспектів його поведінки під час виконання. Наприклад, можна враховувати частоту викликів функцій чи інтенсивність обміну даними між модулями, що впливає на тестову проникність у реальних умовах експлуатації. Такий підхід дозволяє оцінити, як поведінка системи під навантаженням ускладнює доступ тестів до критичних компонентів, наприклад, у розподілених системах чи мікросервісах, де мережеві затримки чи асинхронність додають нові бар'єри [5]. Для цього ІТП може бути доповнений параметром "динамічна проникність", який обчислюється на основі логів виконання або профілювання коду. Це дає змогу адаптувати метод до сучасних архітектур, таких як хмарні обчислення, де традиційні статичні метрики можуть бути недостатніми.

Практична реалізація ІТП із урахуванням динамічних аспектів потребує інтеграції з інструментами моніторингу продуктивності, такими як Jaeger чи Prometheus, які вже широко застосовуються для відстеження поведінки систем [6]. Наприклад, аналіз трасування викликів може виявити "вузькі місця" в архітектурі, де тести не досягають компонентів через високу затримку чи складні умови виконання.

Впровадження ІТП змінює підходи до тестування, переносячи фокус із традиційних метрик на структурну проникність:

- **Раннє виявлення проблем**: Оцінка тестованості на етапі проєктування дозволяє зменшити щільність залежностей чи глибину архітектури.
- **Оптимізація тестових стратегій**: Зосередження зусиль на компонентах із низькою проникністю підвищує ефективність тестування.
- **Покращення якості тестів**: Низький ІТП сигналізує про потребу в складніших тестах для охоплення внутрішніх компонентів.
- **Автоматизація**: Інтеграція ІТП в інструменти розробки (IDE, CI/CD) забезпечує зворотний зв'язок у реальному часі.
- **Зміна пріоритетів**: Архітектурна тестованість стає критерієм якості проєкту.

Додатковий вплив ІТП проявлятиметься в адаптації стратегій тестування: у монолітах він вказуватиме на рефакторинг через залежності, у мікросервісах — на тестування взаємодій. ІТП оцінюватиме вплив змін у коді на тестованість в Agile [4].

Висновки. "Індекс тестової проникності" є підходом до оцінки тестованості, що фокусується на структурній проникності тестів, сприяючи створенню надійніших програмних систем. Він дозволяє прогнозувати проблеми тестування ще на етапі проєктування, оптимізувати розподіл ресурсів і підвищувати якість тестів, що особливо важливо для складних систем із глибокими залежностями [5]. ІТП також відкриває можливості для автоматизації аналізу тестованості, інтегруючись у сучасні процеси розробки, такі як CI/CD [6]. У порівнянні з традиційними метриками, як-от покриття коду, ІТП надає цілісніший погляд на тестованість, враховуючи архітектурні аспекти, що раніше залишалися поза увагою [1]. Подальші перспективи включають його адаптацію до систем із високим рівнем паралелізму, де проникність тестів може залежати від конкурентних потоків виконання, а також інтеграцію з технологіями штучного інтелекту для автоматичного генерування тестових сценаріїв на основі аналізу ІТП [3]. Це може значно скоротити час на тестування, одночасно підвищивши його точність і повноту. Крім того, ІТП може стати основою для стандартизації оцінки тестованості в індустрії, впливаючи на розробку нових рекомендацій і практик у сфері інженерії програмного забезпечення. Подальші дослідження можуть включати емпіричну валідацію ІТП на реальних проєктах і розробку інструментів для його практичного застосування.

Список літератури

1. Alenezi M. Investigating Software Testability and Test Cases Effectiveness // International Journal of Advanced Computer Science and Applications (IJACSA). — 2022. — Vol. 13, No. 1. — P. 629–635. — DOI: 10.48550/arXiv.2201.10090. — Режим доступу: https://www.researchgate.net/publication/358117517 _Investigating_Software_Testability_and_Test_cases_Effectiveness.

2. Starov O., Vilkomir S., Gorbenko A., Kharchenko V. Testing-as-a-Service for Mobile Applications: State-of-the-Art Survey // Dependability Problems of Complex Information Systems / ed. by W. Zamojski, J. Sugier. — Cham : Springer, 2015. — P. 55–71. — (Advances in Intelligent Systems and Computing, vol. 307). — DOI: 10.1007/978-3-319-08964-5_4. — Режим доступу: https://www.researchgate.net/publication /278681802_ Testing-as-a-Service_ for_Mobile_Applications_State-of-the-Art_Survey.

3. Akila V., Vasuki A., Christaline J. A., Sathiya R., Rishi P., Edward A. S. Enhancing Software Testing with Machine Learning Techniques // 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 23–25 March 2023, Erode, India. — Piscataway, NJ : IEEE, 2023. — P. 112–118. — DOI: 10.1109/ICSCDS56580.2023.10105028. — Режим доступу: https://ieeexplore.ieee.org/ document /10105028.

4. Beyer D. Status Report on Software Testing: Test-Comp 2021 // Fundamental Approaches to Software Engineering, 24th International Conference, FASE 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 – April 1, 2021, Proceedings / ed. by E. Guerra, M. Stoelinga. — Cham : Springer, 2021. — P. 341–357. — (Lecture Notes in Computer Science, vol. 12649). — DOI: 10.1007/978-3-030-71500-7_17. — Режим доступу: https://www.researchgate.net/publication/350194257_ Status_Report_on_Software_Testing_Test-Comp_2021

5. Uchôa A., Vieira da Silva C. B., Oizumi W., Blenilio P., Lima R., Garcia A., Bezerra C. I. M. How Does Modern Code Review Impact Software Design Degradation? An In-depth Empirical Study // 36th International Conference on Software Maintenance and Evolution (ICSME), 27 September – 3 October 2020, Adelaide, Australia. — Piscataway, NJ : IEEE, 2020. — P. 511–522. — DOI: 10.1109/ICSME46990.2020.00055. — Режим доступу: https://www.researchgate.net/publication/
343862772_How_Does_Modern_Code_Review_Impact_Software_Design_Degradation_An_In-depth_
Empirical_Study.

6. Charoenwet W., Thongtanunam P., Pham V., Treude C. Toward Effective Secure Code Reviews: An Empirical Study of Security-Related Coding Weaknesses // Empirical Software Engineering. — 2024. — Vol. 29, No. 4. — Article 89. — DOI: 10.1007/s10664-024-10496-y. — Режим доступу: https://link.springer.com/article/10.1007/ s10664-024-10496-y

УДК 004.4

Nina Zdolbitska, Ph.D., Bohdan Heiko, Yuliia Mitsura
ninazdolb@gmail.com, geikobogdanv@gmail.com, yuliamicyra11@gmail.com
Lutsk National Technical University, Lutsk

CLIENT-SERVER APPLICATION FOR PLANNING AND PRODUCTION SCHEDULING BASED ON WPF

Production planning and sequencing are the core of manufacturing companies' performance, and we will evaluate the current state of such research. It is necessary to take into account the constantly changing and evolving market demands, which in turn makes the production process complex. During the production process, companies must use as few resources as possible to ensure high product quality and respond quickly to market demands. Therefore, the need for effective production planning, scheduling, and sequencing has been a very important area of development for companies and researchers in recent decades [1].

In recent years, the Industry 4.0 paradigm has gained momentum, which aims to explore the capabilities of this technology in operations and production management, especially in production planning, planning, execution and control [2]. Industry 4.0 technologies create a connected information system that allows for real-time control, monitoring, and optimization. An integrated production planning and scheduling system is necessary to meet the needs of high-quality production. Most companies, in order to improve the productivity of their manufacturing systems, develop programs to enhance and accelerate the lead time, aimed at reducing machine downtime, increasing operator skills or machine production capacity, reducing defective products, etc.

Effective decision-making tools, such as through cloud services, allow manufacturing managers to combine effective production management strategies and daily control of operations to facilitate decision-making [3]. The process of demand and sales forecasting depends on real-time production planning and supply chain management and is used to understand and take into account customer needs and expectations.

Implementing a client-server application for real-time production planning and scheduling in an industrial company has the potential benefits of planning and controlling production stages, facilitating the process, and reducing complexity and risks.

Creating a client-server application for planning and scheduling production based on WPF is a necessary, interesting, and complex task. Let's describe the main stages that you should pay attention to when creating such an application:

1) The application architecture includes:

- the client part (WPF) – the user interface, where data is displayed, the ability to interact with graphs, input forms, etc.;
- the server part: API for processing requests from the client, data management, and business logic;

2) Technologies:

- WPF (Windows Presentation Foundation) for implementing an attractive user interface;
- Transact-SQL (T-SQL) – a procedural extension of the SQL language;
- SQL Server for data storage.

3) Development of the project structure and folders for organizing the code;

4) Interaction between the client and the server, using HTTP requests for data exchange;

5) Testing the quality of the code by conducting unit tests;

6) Deployment.

This project requires knowledge in various development areas, such as programming, database design, and experience with WPF. However, WPF provides opportunities for developers: through integration with Visual Studio, WPF development becomes more convenient with design and debugging tools.

References

1. Guzman E., Andres B., Poler R. Models and algorithms for production planning, scheduling and sequencing problems: A holistic framework and a systematic review. Journal of Industrial Information Integration, 27, 2022. p. 100287.

2. Guo D., Li M., Zhong R., Huang, G.Q. Graduation Intelligent Manufacturing System (GiMS): an Idustry 4.0 paradigm for production and operations management. Industrial Management & Data Systems, 121(1), 2021. Pp. 86-98.

3. Zdolbitska N., Komenda T., Terletskyi T., Ostapchuk O. Information System of Management and Decision-Making Using Fuzzy Logic. IX International scientific and practical conference «Scientific Problems and Options for Their Solution» (February 7-9, 2024) Bucharest, Romania, International Scientific Unity, 2024. Pp. 96-98.

УДК 004.8

С.А. Коротенко
аспірант, s.korotenko@e-u.edu.ua
Науковий керівник: Р. О. Яровий, *к.т.н., доцент*
декан факультету інформаційних систем та технологій
ПВНЗ «Європейський університет», Київ

ПРЕДСТАВЛЕННЯ BPMN-СТРУКТУРИ БІЗНЕС-ПРОЦЕСУ У ФОРМАТІ ГРАФА ДЛЯ GNN: ВИКЛИКИ І РІШЕННЯ

Постановка проблеми

Більшість існуючих підходів спираються лише на логи подій, ігноруючи структурну інформацію моделі процесу в нотації BPMN (Business Process Model and Notation), яка визначає дозволені переходи, типи задач та бізнес-логіку. Попри графову природу BPMN, її представлення у форматі, придатному для графових нейронних мереж (GNN), залишається малодослідженим. Це обмежує здатність моделей точно враховувати як контекст, так і можливі варіанти розвитку процесу.

Аналіз літератури

Аналіз останніх досліджень показує, що у process mining більшість підходів орієнтовані на обробку логів подій (наприклад, LSTM, RNN). Роботи які залучають графові нейронні мережі, часто обмежуються лінійними послідовностями дій. Наприклад, у роботі Hanga et al., 2020 запропоновано комбінацію LSTM з графовою візуалізацією для пояснення прогнозів, однак без прямого подання BPMN-структури [1]. У дослідженні Sommers et al., 2021 запропоновано використовувати GNN для перетворення логів у Petri-мережі, що є кроком у бік структурного навчання, але без підтримки нотації BPMN [3]. Згідно з Kalenkova et al., 2017, перетворення логів у BPMN-моделі поєднує контрольний і поведінковий рівні [2], однак безпосереднє використання BPMN-структур у GNN досі залишається малодослідженим.

Цілі та завдання

Представити підхід до трансформації BPMN-моделі у граф з ознаками для подальшого використання у GNN з метою прогнозування наступних дій і часу їх виконання.

Виклад основного матеріалу

У сучасних задачах інтелектуального аналізу процесів дедалі більшої актуальності набуває не лише прогнозування наступних дій у бізнес-процесах, а й ефективне представлення самих процесів у вигляді структур, придатних до аналізу нейронними мережами. Одним з найперспективніших напрямів є використання GNN, які здатні враховувати як локальні залежності між елементами процесу, так і глобальний контекст. Ключова задача — формування графової репрезентації моделі бізнес-процесу, зокрема заданої у нотації BPMN, для подачі на вхід GNN разом з логами подій.

BPMN — стандарт формалізованого опису бізнес-процесів у вигляді графа з вузлами (події, задачі, шлюзи) та ребрами (послідовності дій). Щоб адаптувати такі моделі до аналізу через GNN, необхідно трансформувати BPMN-діаграму у формат, сумісний із графовими структурами бібліотек машинного навчання, наприклад PyTorch Geometric.

Підхід передбачає рекурсивне перетворення BPMN-діаграми у графову структуру шляхом обходу елементів у форматі XML. Для кожного вузла BPMN (наприклад, userTask, serviceTask, exclusiveGateway, parallelGateway, startEvent, endEvent) визначається тип та зв'язки з іншими вузлами згідно з послідовностями. Якщо вузол є зовнішнім викликом (наприклад, callActivity), відбувається рекурсивне підвантаження відповідного підпроцесу з іншого BPMN-файлу. В результаті будується єдиний зв'язний граф, який охоплює всі рівні вкладеності процесу. До кожного вузла додаються фічі, збагачені даними з логів виконання: часові мітки, статус, тривалість, кількість виконань, черговість виконання тощо. Побудований граф включає матрицю ознак вузлів $\boldsymbol{x}$, індекси зв'язків edge_index, атрибути ребер та, за потреби, вектор ознак документа або мітки для задачі прогнозу.

Основною складністю при перетворенні BPMN-моделі у формат, придатний для графових нейронних мереж, є гетерогенність отриманого графа. Вузли бізнес-процесу можуть представляти різні семантичні типи: користувацькі задачі (userTask), автоматизовані сервіси (serviceTask), події (startEvent, endEvent), умовні (exclusiveGateway) та паралельні шлюзи (parallelGateway). Кожен тип має власний набір атрибутів, що ускладнює формування однорідного простору ознак. Для коректної обробки в GNN усі вузли мають бути представлені у вигляді векторів однакової розмірності. Це забезпечується фічеризацією: числові ознаки (тривалість, кількість виконань, часові мітки) включаються напряму, а категоріальні (тип вузла, статус, код дії) перетворюються за допомогою one-hot або векторних ембедінгів. Аналогічна уніфікація застосовується до ребер, якщо ті мають атрибути (наприклад, тривалість переходу або логічний тип зв'язку). Такий підхід дозволяє сформувати батчі з графів у форматі x, edge_index, edge_attr та використовувати BPMN як повноцінне джерело структурованих вхідних даних для моделей GNN, здатних обробляти гетерогенні графи.

Другим важливим викликом є варіативність структури процесу: одна й та сама бізнес-задача може мати різні контексти виконання — залежно від типу кейсу, організаційної ролі, року чи регіону. Це призводить до

утворення множини варіантів графів для однієї логіки процесу, що ускладнює навчання моделей машинного навчання через зміну розподілу структур. Зокрема, це спричиняє зниження узагальнюваності моделі на нових даних (distribution shift). Для формування стабільного і репрезентативного датасету необхідно проводити нормалізацію графів: виключати рідкісні або нестабільні вузли, групувати семантично подібні задачі, узгоджувати формат подання та зменшувати кількість топологічних варіантів.

Третім викликом є підтримка паралельних та умовних гілок виконання, які є характерними для BPMN-моделей. У таких випадках важливо не лише відслідковувати історію вже виконаних задач, а й зберігати інформацію про потенційні варіанти продовження — тобто про суміжні вузли, які ще не були активовані, але структурно доступні. Втрата цієї інформації призводить до неповного розуміння контексту. Для вирішення цієї проблеми пропонується подавати на вхід GNN повну топологію процесу, з маркуванням активних вузлів, що дозволяє моделі обробляти як історичну траєкторію, так і можливі сценарії подальшого виконання. Такий підхід забезпечує контекстуалізоване подання графа, подібно до attention-масок у трансформерах.

На практиці, на вхід моделі GNN подається повний граф процесу, визначається як $G = (V, E)$, де V — множина вузлів, E— множина зв'язків між ними. Кожен вузол має вектор ознак x_v, а ребро — атрибут e_{ij}. $V' \subset V$ — підмножина вже виконаних вузлів. Задача прогнозу наступної дії формулюється як багатокласова класифікація:

$$\hat{y} = \text{argmax}_y P(y \mid G, V')$$

а прогноз часу її виконання — як регресія:

$$\hat{t} = f(G, V')$$

Відповідна архітектура показана на рис.1.

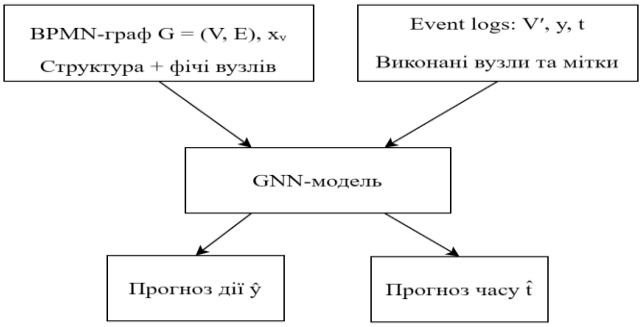


Рис. 1. Архітектура моделі прогнозування на основі GNN для BPMN-процесів

Висновки

Запропонований підхід забезпечує ефективну адаптацію BPMN-моделей для аналізу графовими нейронними мережами, зберігаючи семантику процесу та інтегруючи дані виконання.

Подальші дослідження мають бути спрямовані на:

- емпіричне порівняння з моделями, що працюють лише на логах;
- вивчення впливу типу фічеризації на точність прогнозу;
- дослідження узагальнення моделі на процесах з частковою або змінною структурою.

Список літератури:

1. Hanga K. M., Kovalchuk Y., Gaber M. M. A Graph-Based Approach to Interpreting Recurrent Neural Networks in Process Mining. *IEEE Access*. Vol. 8, 2020. P. 172923–172938. DOI:10.1109/ACCESS.2020.3025999.

2. Kalenkova A. A., Aalst W. M. P. van der, Lomazova I. A. et al. Process mining using BPMN: relating event logs and process models. *Software & Systems Modeling*. Vol. 16, Issue 4. P. 1019–1048. DOI:10.1007/s10270-015-0502-0.

3. Sommers D., Menkovski V., Fahland D. Process Discovery Using Graph Neural Networks. 13.09.2021. URL: http://arxiv.org/abs/2109.05835

УДК 004.4

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

ТЕОРІЯ ІГОР ЯК ІНСТРУМЕНТ ОПТИМІЗАЦІЇ ВЗАЄМОДІЇ МІКРОСЕРВІСІВ

Мікросервісна архітектура (МСА) стала фундаментом побудови масштабованих та динамічних програмних систем. Вона забезпечує модульність, незалежність сервісів та гнучкість у розгортанні, однак водночас створює виклики щодо балансування навантаження, ефективного розподілу ресурсів і підтримання узгодженої якості обслуговування (QoS) [1]. Традиційні підходи до керування ресурсами у хмарі (оркестратори на кшталт Kubernetes) використовують наперед задані евристики балансування, як-от Binpack чи Spread у Docker Swarm, які не завжди гарантують оптимальний розподіл ресурсів між всіма сервісами. Зі збільшенням складності взаємозалежностей мікросервісів та динаміки навантаження виникає потреба у нових методах, що забезпечать оптимальне розподілення ресурсів, масштабування і надійність системи в режимі реального часу.

Теорія ігор пропонує формальний підхід до моделювання взаємодії мікросервісів як автономних агентів із конфліктними або спільними цілями [2]. У контексті МСА це означає, що кожен мікросервіс можна розглядати як гравця, який прагне покращити власні показники (наприклад, мінімізувати час відповіді або максимально використовувати виділені ресурси). Взаємодію мікросервісів, як конкуренцію за ресурси чи співробітництво заради досягнення спільної мети можна описати через відповідні ігрові моделі. В такій моделі кожен сервіс виступає гравцем, який обирає стратегію використання ресурсів, зважаючи на рішення інших. Це дозволяє аналізувати не лише індивідуальну ефективність, а й глобальні наслідки децентралізованого прийняття рішень. Застосування кооперативних та некооперативних моделей гри дає змогу виявити оптимальні стратегії взаємодії, що мінімізують затримки, забезпечують справедливий розподіл обчислювальних потужностей та підвищують стійкість системи в умовах змінного навантаження. Некооперативні моделі, зокрема ігри типу congestion games, дозволяють виявити рівновагу Неша у розподілі обчислювальних ресурсів між сервісами [3]. Така рівновага відображає ситуацію, за якої жоден сервіс не має стимулу змінити свою стратегію односторонньо, що забезпечує стабільність системи. Водночас кооперативні ігри, як-от моделі на основі Nash Bargaining, спрямовані на досягнення соціально оптимального балансу між сервісами з урахуванням глобальної мети [4].

Сучасні дослідження демонструють ефективність теоретико-ігрових методів у різних аспектах МСА: від керування ресурсами і балансування навантаження до забезпечення безпеки та оптимізації на периферії мережі. Вони підтверджують, що застосування ігрових стратегій у керуванні МСА дозволяє зменшити середній час відповіді, підвищити утилізацію ресурсів і знизити частоту відмов за умов пікового навантаження [5]. Окрім суто ігрових підходів, варто згадати й про інтеграцію з методами машинного навчання. Зокрема, дедалі частіше застосовуються алгоритми глибокого підкріпленого навчання (reinforcement learning, RL) для навчання оптимальних стратегій керування мікросервісами. Крім того, поєднання теорії ігор з машинним навчанням дозволяє реалізувати адаптивне управління, в якому стратегії мікросервісів змінюються динамічно на основі прогнозів навантаження.

Таким чином, інтеграція підходів теорії ігор у процес управління мікросервісною архітектурою відкриває нові можливості для побудови самонавчальних і адаптивних систем з високими показниками продуктивності та надійності. Теоретико-ігровий підхід формує нову парадигму в оптимізації мікросервісних систем, де автономія сервісів узгоджується із загальною ефективністю системи.

Список літератури

1. G. D. Maayan, "5 Ways to Reduce the Attack Surface for Microservices," [Online]. Available: https://securityboulevard.com/2023/04/5-ways-to-reduce-the-attack-surface-for-microservices.

2. J. Johnson and L. Shiff, "What Is Microservice Architecture? Microservices Explained," [Online]. Available: https://www.bmc.com/blogs/microservices-architecture.

3. "8 Ways to Secure Your Microservices Architecture," [Online]. Available: https://www.okta.com/resources/whitepaper/8-ways-to-secure-your-microservices-architecture.

4. C. Walia, Mastering Cloud-Native Microservices: Designing and Implementing Cloud-Native Microservices for Next-Gen Apps, BPB Publications, 2023, p. 298.

5. S. Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly, 2021, p. 6.

УДК 004.9

М.О. Рисований, *магістрант*
maxcimofficial@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

ПЕРЕВАГИ ТА НЕДОЛІКИ РОЗРОБКИ ІГОР ТА ПЗ НА UNREAL ENGINE

Unreal Engine – це потужний багатоплатформний ігровий рушій, розроблений компанією Epic Games, призначений для створення комп'ютерних ігор, інтерактивних додатків та симуляцій. Рушій підтримує розробку застосунків для широкого спектра платформ, включаючи персональні комп'ютери (Windows, macOS, Linux), ігрові консолі (PlayStation, Xbox, Nintendo Switch), мобільні пристрої (Android, iOS) та системи віртуальної реальності. На основі Unreal Engine створено тисячі ігор та інтерактивних проєктів різних жанрів. Рушій здобув особливу популярність серед великих студій розробки, які використовують його для високобюджетних AAA-проєктів із передовою графікою, а також знайшов застосування поза ігровою індустрією – зокрема, в архітектурній візуалізації та кіновиробництві. Варто зазначити, що Unreal Engine доступний не тільки глобальним компаніям, а й незалежним розробникам: базова версія рушія є безкоштовною для користування.

Основною перевагою Unreal Engine є неперевершена якість графіки та потужні засоби для реалізації реалістичних віртуальних світів. Рушій забезпечує сучасні технології рендерингу з підтримкою передового освітлення і тіней, систем частинок та інших візуальних ефектів, що дає змогу досягти фотореалістичного рівня зображення. Unreal Engine орієнтований на створення масштабних сцін із високою деталізацією, складною фізикою та анімацією, які відповідають вимогам AAA-ігор. Саме тому Unreal Engine став одним з головних стандартів індустрії для проектів AAA-рівня, у яких демонструються найновіші досягнення в графіці.

Наступною перевагою є наявність вбудованої системи візуального скриптування Blueprints. Unreal Engine містить інтуїтивний інтерфейс на основі блок-схем (нодів), що дозволяє створювати ігрову логіку без написання традиційного коду. Це спрощує прототипування ігор та дає можливість дизайнерам і художникам налаштовувати поведінку об'єктів без глибоких знань мов програмування.

Unreal Engine забезпечує розробників комплексним набором інструментів «із коробки». Рушій має потужний вбудований фізичний двигун, систему анімації та поведінки персонажів (штучного інтелекту), підтримку мережевої багатокористувацької взаємодії, а також засоби для розробки проєктів віртуальної і доповненої реальності. Така різноманітність можливостей дозволяє вирішувати більшість типових завдань без залучення сторонніх бібліотек, що прискорює процес розробки та полегшує створення складних ігор. Epic Games постійно оновлює Unreal Engine, додаючи новітні технології.

Недоліком Unreal Engine можна вважати його ресурсоємність та потребу у потужному обладнанні. Рушій доволі **«важкий»**: проекти на Unreal займають значний обсяг пам'яті і дискового простору, а сам редактор для комфортної роботи вимагає продуктивного процесора та графічної карти. Використання повного потенціалу рушія супроводжується високими системними вимогами. Це стосується як етапу розробки (на слабких машинах середовище Unreal може працювати повільно), так і кінцевих користувачів – готова гра на UE часто потребує більш потужного «заліза» для плавної роботи. Без належної оптимізації існує ризик проблем з продуктивністю на мобільних пристроях або менш потужних ПК, що накладає додаткові зусилля на розробників при адаптації проекту під такі платформи.

Отже, Unreal Engine є **потужним середовищем розробки**, що надає сучасні інструменти для створення високоякісних інтерактивних продуктів. Незважаючи на окремі недоліки (високі вимоги до обладнання, складність навчання та модель роялті), Unreal Engine залишається одним з провідних виборів для розробників, особливо коли цілі проекту включають найвищу якість графіки та застосування передових технологій. Доцільність використання цього рушія залежить від поставлених завдань: якщо проект вимагає фотореалістичної 3D-графіки, складної фізики чи орієнтований на рівень AAA з відповідними ресурсами – Unreal Engine є виправданим рішенням. Водночас, для невеликих за масштабом ігор або обмежених за ресурсами команд альтернативні, більш легкі рушії можуть виявитись практичнішими.

Список літератури

1. Aram Cookson, Ryan DowlingSoka, Clinton Crumpler. *Unreal Engine 4 Game Development in 24 Hours, Sams Teach Yourself*, 2016.
2. Tom Shannon. *Unreal Engine 4 for Design Visualization: Developing Stunning Interactive Visualizations, Animations, and Renderings*, Addison-Wesley, 2017.
3. Brenden Sewell. *Blueprints Visual Scripting for Unreal Engine: Build professional 3D games with Unreal Engine 4's Visual Scripting system*, Packt Publishing, 2015.

УДК 004.8

Л.І.Поліщук
старший викладач, pli_80@ukr.net
Центральноукраїнський національний технічний університет, Кропивницький

ВИКОРИСТАННЯ ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ В ПРОГРАМУВАННІ

Штучний інтелект (ШІ) для створення тестів стає все більш популярним інструментом у галузі освіти. Завдяки здатності автоматизувати процес створення, оцінки та аналізу тестів, ШІ може значно полегшити життя викладачів і забезпечити більш персоналізоване навчання для здобувачів. Слід виділити такі найкращі генератори тестів, як QuizGecko, Quizlet, Qzzr, Riddle.com, Playbuzz, Socrative, Edmodo, Conker, QuizGecko, Revisely. Усі інструменти на основі ШІ мають свої особливості, але важливо зауважити, що їхнім основним недоліком є те, що більшість з них орієнтовані на англомовних користувачів і безкоштовні сервіси обмежені. Це може створювати певні труднощі при використанні. Тому командою «На Урок» було створено інший інструмент на базі ШІ - «Персональний помічник сучасного вчителя» [1]. Для роботи інструмента потрібен стабільний доступ до інтернету та базові технічні навички, що може бути проблемою за деяких умов.

Для рішення цієї проблеми і було проведено дослідження, основною метою якого є автономність застосування штучного інтелекту для автоматизації процесу створення тестів. У ході роботи використовувався ШІ Ollama для генерації тестових питань. Цей інструмент здатний аналізувати тематику і створювати запитання відповідно до заданих параметрів, що дозволяє значно зекономити час та забезпечити збереження великої кількості інформації для подальшого використання.

Ollama AI підтримує роботу з відкритими моделями, які можна завантажувати та запускати через командний рядок. Великі мовні моделі (LLM), які використовуються Ollama, дозволяють виконувати завдання генерації й перекладу тексту, забезпечуючи якість, що наближається до рівня природної мови. Ці моделі навчаються на великому обсязі текстових даних і застосовуються для різних задач, включаючи відповіді на запитання, узагальнення, переклади, генерацію зв'язного контенту, завершення тексту та пошук.

Для інсталяції Ollama слід використовувати офіційний сайт [2]. Запуск моделі здійснюється командою *ollama run llama3.1:8b* з командного рядка. Щоб переглянути інформацію про встановлену модель, скористаємось командою *show info* (рис.1). При першому запуску модель завантажується на комп'ютер, і варто враховувати за інформацією [3], що Llama 3.1 8B займає 4.7GB дискового простору.

```
>>> /show info
  Model
        arch                    llama
        parameters              8.0B
        quantization            Q4_0
        context length          131072
        embedding length        4096

  Parameters
        stop    "<|start_header_id|>"
        stop    "<|end_header_id|>"
        stop    "<|eot_id|>"

  License
        LLAMA 3.1 COMMUNITY LICENSE AGREEMENT
        Llama 3.1 Version Release Date: July 23, 2024

>>> Send a message (/? for help)
```

Рис. 1. Інформація про модель

Особливість бібліотеки моделей Ollama — це можливість використання так званих «без цензури» моделей. На відміну від багатьох стандартних моделей ШІ, які обмежують відповіді для забезпечення політичної коректності або відповідності законодавству, такі моделі спрямовані на видалення упередженості й забезпечення точності відповідей, надаючи альтернативні рішення з відкритим кодом.

Приклад використання бібліотеки Ollama в C++ для аналізу тексту на ключові слова:

```
#include "ollama.hpp"
#include <iostream>
int main() {
   std::string текст = " Штучний інтелект змінює світ.";
   std::cout << "Ключові слова: " << ollama::analyze("llama3:8b", текст) << std::endl;
   return 0;
}
```

Результат роботи наведеної програми – це ключові слова: "штучний інтелект, світ, зміни."

Для створення графічного інтерфейсу користувача було використано кросплатформову бібліотеку Qt. Програми, написані із застосуванням Qt, можуть без переробки працювати в середовищі сучасних операційних

систем Windows, Linux, macOS, Android і iOS. Саме Ollama, використовується в поєднанні з Qt для створення інтерактивних додатків, які обробляють запити користувача та взаємодіють з цією моделю.

Офіційний веб-сайт Qt [3] дає можливість завантажити останню версію.

Було реалізовано тестування математичних завдань різної складності (рис.2), де штучний інтелект генерує задачі (рис.3), враховуючи параметри складності, задані користувачем. Додаток дозволяє отримувати миттєвий зворотний зв'язок, оскільки після кожної відповіді система автоматично генерує повідомлення про її правильність (True/False). Крім автоматичного підрахунку позитивних балів із загальної кількості (рис.4), була впроваджена функція аналізу типових помилок, яка допомагає виявляти слабкі місця у знаннях.

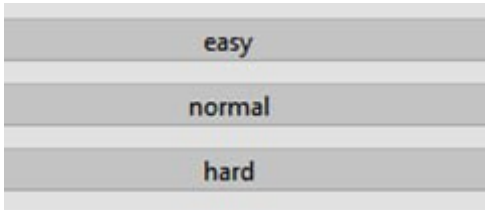


Рис. 2. Вибір різної складності задач

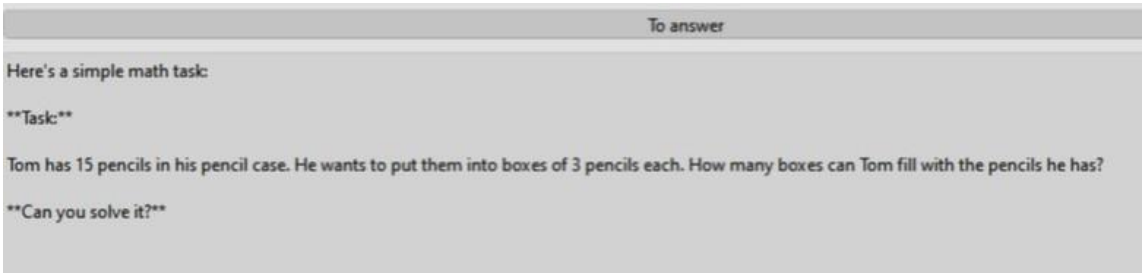


Рис. 3. Згенерована задача

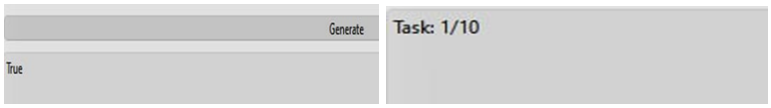


Рис. 4. Результат тестування

Інтерфейс тестування передбачає можливість перегляду історії результатів, що забезпечує оцінку прогресу користувача протягом усіх тестів. Також інтегровано систему підказок для складних задач, яка може надавати рекомендації щодо їх вирішення. Це робить процес навчання не тільки ефективним, але й цікавим та інтерактивним, сприяючи мотивації до подальшого вдосконалення навичок.

Застосування штучного інтелекту у процесі програмування, зокрема для автоматизації створення тестів, відкриває нові горизонти в освітньому процесі. Вибір інструментів на базі ШІ, таких як Ollama, дозволяє значно полегшити викладачам задачу з розробки тестових питань, одночасно забезпечуючи збереження високої якості та точності контенту. У поєднанні з потужними бібліотеками для створення графічних інтерфейсів, такими як Qt, стало можливим створення інтерактивних програм, які не тільки генерують задачі, але й надають зворотний зв'язок, аналізують помилки та допомагають у вдосконаленні навчальних навичок.

Незважаючи на значні переваги, такі як автономність роботи, можливість налаштування моделей під специфічні потреби та економія на хмарних сервісах, є й певні недоліки, зокрема необхідність у потужному обладнанні та технічному обслуговуванні. Окрім того, проблемою для деяких користувачів може бути відсутність зручності у використанні таких інструментів, як Ollama, у випадку обмеженого доступу до інтернету.

Проте, потенціал використання ШІ в освіті, зокрема у генерації тестових завдань та аналізі результатів, не викликає сумнівів.

Список літератури

1. Створення тестів за допомогою штучного інтелекту: https://naurok.com.ua/post/stvorennya-testiv-za-dopomogoyu-shtuchnogo-intelektu (дата звернення: 08.04.2025)

2. Download Ollama. URL: https://ollama.com/download (дата звернення: 07.04.2025) https://naurok.com.ua/post/stvorennya-testiv-za-dopomogoyu-shtuchnogo-intelektu

3. Model library. URL: https://github.com/ollama/ollama/blob/main/README.md (дата звернення: 08.04.2025)

4. Qt. URL: Офіційний веб-сайт Qt: https://www.qt.io/ (дата звернення: 07.04.2025)

УДК 004.42:004.89

О.К. Конопліцька-Слободенюк, *викладач*, А.Я. Клюй *студенка 2-го курсу*
ksuha80@gmail, nastya.klui@gmail.com
Центральноукраїнський національний технічний університет,. Кропивницький

ОГЛЯД ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ З БАЗАМИ ДАНИХ

Зараз у базах даних (БД) зберігається багато інформації, і з кожним роком її стає ще більше. Через це звичайні методи керування не завжди справляються. Тому все частіше використовують інструменти зі штучним інтелектом (ШІ). Вони допомагають автоматизувати рутинні процеси, краще аналізувати дані, швидше працювати з базою та зручніше взаємодіяти з нею. *Метою* аналізу є вивчення сучасних інструментів на основі ШІ для покращення функціональних можливостей та ефективності роботи з БД, що робить тему огляду актуальною та вартою уваги. *Об'єктом* є процес керування БД із використанням ШІ-інструментів. *Предметом* — засоби ШІ, що оптимізують обробку, зберігання та аналіз даних у БД. Щоб досягти мети, було окреслено такі завдання: визначити ключові інструменти ШІ для роботи з БД; класифікувати їх за напрямами використання; дослідити функціональні можливості кожного інструмента; порівняти переваги та недоліки; оцінити їхній вплив на продуктивність і керованість БД.

Після опрацювання технічних джерел [1–4, 7] та публікацій [5, 6] вдалося визначити кілька напрямів, у яких найчастіше використовують ШІ-інструменти при роботі з БД. Ці сфери вирізняються тим, що допомагають працювати з великим обсягом інформації швидко, зручно та з мінімальними затратами. Кожна з груп інструментів має своє призначення. Їх можна умовно поділити так: **Інструменти для роботи з неструктурованими даними** (Towhee, Zilliz); **Навчання моделей у межах БД** (PostgresML, MindsDB); **Пошук схожих даних у великих масивах** (Pinecone, Qdrant); Візуальне проєктування БД (Workik AI, Taskade AI); **Хмарні платформи з простими інтерфейсами** (MongoDB Atlas, Airtable AI).

Серед ключових інструментів найбільше виділяються Zilliz (на базі Milvus) для обробки великих обсягів неструктурованих даних, а також Qdrant, що в поєднанні з мовними моделями забезпечує швидкий пошук схожих записів [5, 6]. PostgresML [5] і MindsDB [3, 5] дають змогу створювати моделі прогнозування за допомогою звичних SQL-запитів, тоді як Towhee [5, 6] використовується для попередньої обробки даних. Airtable AI [1, 5] забезпечує зручну роботу з таблицями, Workik AI [5–7] та Taskade AI [5, 6] – візуальне проєктування БД з генерацією API, а MongoDB Atlas [3, 5] вирізняється підтримкою аналітики в режимі реального часу та масштабованістю.

На основі проаналізованих джерел [1–7] було складено узагальнену власну Таблицю 1 — «Огляд інструментів для підвищення ефективності роботи з базами даних».

Таблиця 1

Огляд сучасних інструментів ШІ для роботи з базами даних

Інструмент	**Призначення**	**Тип даних**	**Інтеграція**	**Переваги**	**Недоліки**
Towhee	Обробка запитів	Неструктуровані	API /SDK	Автоматизація запитів	Обмеження при складних запитах
Zilliz	Векторні дані	Векторні, неструктуровані	Milvus API	Масштабованість, ефективність	Складність, залежність від Milvus
PostgresML	Машинне навчання в БД	Структовані (SQL-дані)	PostgreSQL API/SDK	SQL-сумісність, проста інтеграція	Обмежена підтримка інших СУБД
MindsDB	Машинне навчання із SQL-запитів	Структуровані (SQL-дані)	PostgreSQL API/SDK	Легка інтеграція, швидке навчання моделей	Обмежена гнучкість
Pinecone	Векторний пошук	Векторні	PostgreSQL API/SDK	Продуктивність, зручна інтеграція	Закрита архітектура, висока вартість
Qdrant	Векторний пошук	Векторні	API/SDK	Точність, швидка обробка	Налаштування під задачі

Продовження Таблиці 1

Workik AI	Візуальне проєктування БД	Структуровані (ER-діаграми, схеми)	SQL, NoSQL, API	Автогенерація API, зручно для новачків	Обмежена функціональність
Taskade AI	Візуалізація БД і задач	Структуровані	API, сервіси	Зручний інтерфейс	Не підтримує складні моделі
MongoDB Atlas	Хмарне управління БД + ШІ	Структуровані, неструктуровані	MongoDB API, хмара	Масштабованість, реальний час	Висока вартість при навантаженнях
Airtable AI	Таблиці + автоматизація	Структуровані (табличні дані)	Airtable API	Інтерфейс для новачків	Обмежена гнучкість, не підтримує складні запити

Щоб легше зрозуміти, з якими інструментами працювати в різних випадках (див. Таблиця 1), було вирішено поділити їх на дві умовні групи. Це неформальна, але практично зручна класифікація, яка виявилася корисною для аналізу. До першої групи віднесено сервіси, які працюють безпосередньо з даними — виконують обробку, аналіз або прогнозування. Серед них: Towhee, Zilliz, PostgresML, MindsDB, Pinecone та Qdrant. Вони корисні, коли потрібно не просто зберігати дані, а й отримувати з них певну користь. **Towhee**, наприклад, показав ефективність під час тестування зображень — він ефективно справився з перетворенням картинок у формат, придатний для бази. **Zilliz** краще показав себе в умовах, коли дані змінюються швидко — зокрема, у задачах з потоковою обробкою. Що цікаво, **PostgresML** і **MindsDB** дозволяють будувати моделі прогнозування, використовуючи звичайні SQL-запити. Для тих, хто вже трохи працював з базами, але не є програмістом таке рішення доволі зручне. А ось **Pinecone** і **Qdrant** — працюють інакше. Вони орієнтовані на пошук схожих об'єктів у великих масивах. Тому якщо брати завдання зі схожістю текстів, то там Qdrant спрацює дуже точно.

Друга група об'єднує сервіси, які допомагають навести порядок у структурі бази та зробити роботу з нею зручною. Включили сюди Workik AI, Taskade AI, MongoDB Atlas і Airtable AI. Їхнє завдання — не обробляти дані, а показати, як усе пов'язано між собою. **Workik AI** необхідно використовувати на етапі проєктування, оскільки він зручно допомагає розкласти все по поличках і навіть згенерує базову документацію. Інструмент **Taskade AI,** зі свого боку, зручний для візуалізації зв'язком між даними у вигляді блок-схем, що особливо корисно, коли працюєш не сам, а в команді. **MongoDB Atlas** демонструє ефективність у роботі з неструкторованими даними, а саме у випадках, де обсяг інформації значний, а формат різноманітний. **Airtable AI**, на відміну від інших, орієнтований на користувачів без попереднього досвіду в роботі з БД, оскільки інтерфейс зрозумілий, а функціональність дозволяє швидко організувати необхідну структуру без складного налаштування.

Висновки. У цій роботі проаналізовано, як саме інструменти зі штучним інтелектом можуть допомагати працювати з базами даних швидше та простіше. Було розглянуто десять популярних платформ — серед них Towhee, Zilliz, PostgresML, MindsDB, Pinecone, Qdrant, Workik AI, Taskade AI, MongoDB Atlas та Airtable AI. У кожної з них свої сильні сторони: одні краще справляються з прогнозами, інші — з неструктурованими даними чи зручними інтерфейсами. Щоб навести порядок у всьому цьому, ми згрупували ШІ-інструменти за типами завдань і рівнем складності. Так стало простіше зрозуміти, що для чого краще підходить і як це обирати під конкретну задачу. У результаті можемо сказати: якщо знати, що саме необхідно, штучний інтелект дійсно допоможе — автоматизує рутину, заощадить час і сили в роботі з базами даних. Причому сьогодні багато рішень доступні навіть тим, хто не має досвіду програмування — і це надзвичайно зручно.

Список літератури:

1. Airtable AI. URL: https://www.airtable.com/platform/ai (дата звернення: 08.04.2025).

2. MindsDB. What is it and what is it used for? URL: https://hawatel.com/en/blog/mindsdb-what-is-it-and-what-is-it-used-for/ (дата звернення: 07.04.2025).

3. MongoDB Atlas. URL: https://www.mongodb.com/docs/atlas/ (дата звернення: 07.04.2025).

4. Qdrant vs Pinecone: Vector Databases for AI Apps. URL: https://qdrant.tech/blog/comparing-qdrant-vs-pinecone-vector-databases/ (дата звернення: 08.04.2025).

5. Смірнов О. А., Константинова Л. В., Конопліцька-Слободенюк О. К., Козірова Н. Л., Якименко Н. М., Доренський О. П., Буравченко К. О. Інструменти штучного інтелекту для роботи з базами даних та аналізу даних Кібербезпека: освіта, наука, техніка, 2024, № 26, с. 6–27.

6. Stonebraker M., Зetintemel U. Data Integration: The Current Status and the Way Forward IEEE Data Eng. Bull. 2005. Vol. 28, № 1, p. 3–9.

7. Workik AI. URL: https://workik.com/ai-powered-rest-api-generator (дата звернення: 08.04.2025).

УДК 004.272, 004.738.5

О.В. Ревнюк, О.С. Улічев
o.revnyuk@e-u.edu.ua, askin79@gmail.com
Приватний вищий навчальний заклад "Європейський університет", Київ, Україна

REINFORCEMENT LEARNING ЯК ОПТИМІЗАЦІЯ ДОСТУПУ ДО ОБ'ЄКТІВ НА EDGE ВУЗЛАХ IOT СИСТЕМ

Сучасні IoT-системи вимагають низької затримки, енергоефективності та оптимального використання пропускної здатності при обміні даними між розподіленими сенсорами, шлюзами та хмарними сервісами. Висока залежність IoT-пристроїв від централізованих хмарних обчислень створює значні проблеми, зокрема збільшення затримки, перевантаження магістральних мереж і високе енергоспоживання через часті запити до віддалених серверів. Це особливо критично для додатків реального часу, таких як промисловий моніторинг, автономний транспорт, розумні міста та цифрове здоров'я. Також IoT вузли мають технічні обмеження компонентів: частота процесора, оперативна пам'ять та інші його складові.

Щоб подолати ці проблеми, у розподілених IoT-системах, активно застосовуються технології периферійних обчислень, туманних обчислень та інформаційно-центричних мереж . Ці підходи забезпечують зберігання та обробку даних безпосередньо на вузлах периферійної мережі, таких як IoT-шлюзи (вузли), базові станції та проміжні обчислювальні платформи. Завдяки цьому зменшується кількість звернень до центрального дата-центру, знижуються затримки обробки запитів та оптимізується використання пропускної здатності мережі.

Кешування, а саме ефективна його реалізація в edge-вузлах є одним із ключових механізмів підвищення ефективності IoT-інфраструктури. Воно дозволяє тимчасово зберігати дані сенсорів, агреговану телеметрію та історичні записи на проміжних вузлах, що значно зменшує кількість запитів до сервера. Наприклад, у промислових системах моніторингу кешування може використовуватися для локального аналізу показників освітленості або температури, з передачею лише критичних змін до центрального сервера. В розумних містах edge-кеш може зберігати останні дані про дорожній трафік для швидкого доступу транспортних систем [1].

Розглянемо доступні варіанти управління кешем. Використання традиційних методів та підходів не досить для раціонального та ефективного збереження об'єктів. Статичні алгоритми не є адаптивними до зміни контенту через те, що втрата ефективності у вузлах несе втрату продуктивності системи загалом. Методи машинного навчання можуть оптимізувати кеш для його ефективного використання. Для цього існують різні підходи ML:

- контрольоване навчання (supervised learning),
- неконтрольоване навчання (unsupervised learning),
- навчання з підкріпленням (RL),
- нейронні мережі (NN),
- перенесене навчання (TL).

Враховуючи обмеженість до даних користувачів у периферійних мережах, для оптимізації кешування часто застосовуються моделі машинного навчання, які навчаються без попередніх знань (Reinforcement learning). Застосування навчання з підкріпленням (Reinforcement Learning, RL) дозволяє ефективно вирішити проблему адаптивного управління кешем в умовах динамічних мережевих навантажень і змін користувацьких запитів [2].

На відміну від статичних алгоритмів навчання, RL-підхід не потребує попередньо розмічених даних, агент навчається через взаємодію з середовищем, отримуючи винагороду за прийняття оптимальних рішень. У контексті кешування це означає, що агент може навчитися, які дані варто зберігати в кеші, коли їх оновлювати, і як це робити залежно від змін у поведінці користувачів, доступних ресурсів та пропускної здатності. Такий підхід дозволяє досягати високої ефективності в управлінні кешем навіть у гетерогенних середовищах із великою кількістю edge-вузлів та обмеженими обчислювальними потужностями.

Кешування функціонує за класичною схемою «агент – середовище» (рис. 1). Агентом виступає модуль управління кешем на edge-вузлі, а середовищем – система IoT з динамічними запитами до даних. Агент приймає рішення спираючись на поточний стан середовища – наприклад, типи останніх запитів, частоту звернень до об'єктів, обсяг кешу та мережеву затримку. Після кожної дії агент отримує винагороду, яка відображає ефективність дії. З часом агент навчається максимізувати сумарну довгострокову винагороду, виробляючи політику кешування, яка адаптується до змін у системі. Таким чином, RL дозволяє кешу працювати не лише швидко, а й ефективно – прогнозуючи майбутні запити й підвищуючи загальну ефективність IoT-інфраструктури. Це особливо корисно у випадках, коли характер трафіку змінюється упродовж доби або залежить від зовнішніх чинників (наприклад, зміни погоди, часу доби або подій у місті). Завдяки функції довгострокової винагороди, підкріплювальне навчання дозволяє формувати стратегії, що зменшують кількість промахів кешу (cache misses) і знижують затримку доступу до інформації. Впровадження RL у системи кешування IoT також відкриває можливості до колективного навчання між вузлами, де кожен

агент може локально адаптувати стратегію кешування, а спільне середовище забезпечує узгоджене функціонування всієї мережі.

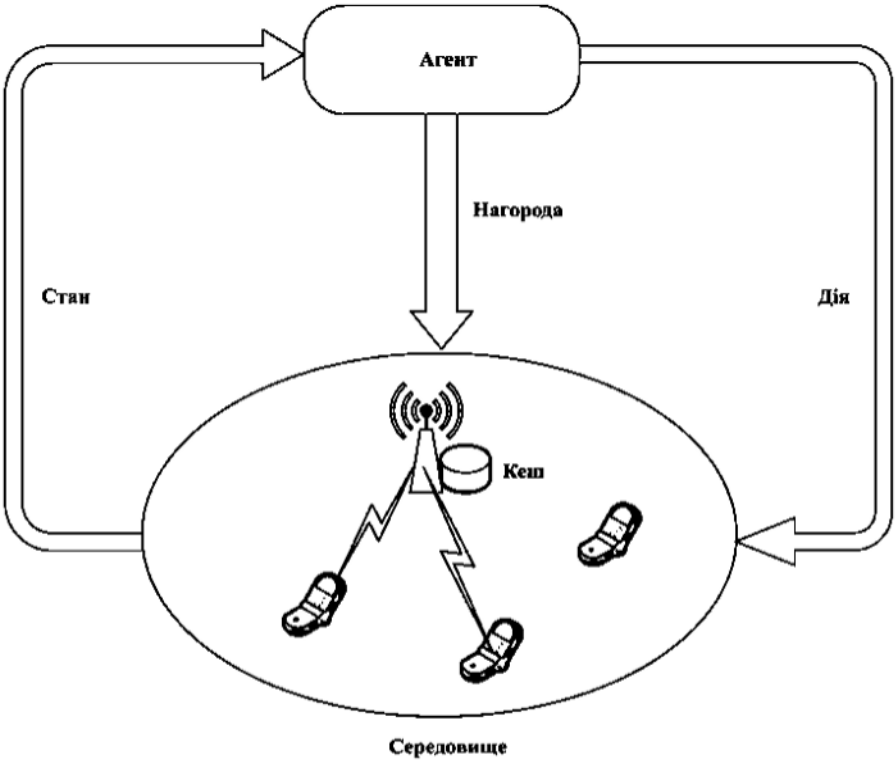


Рис. 1 Схема роботи ефективного кешування

Ефективне кешування в edge-вузлах відіграє важливу роль у зниженні затримок, зменшенні навантаження на мережу та забезпеченні безперебійної роботи IoT-систем у сценаріях з великим потоком даних. Машинне навчання, а саме використання RL алгоритмів є доцільним для ефективного керування збереженням даних.

Список літератури

1. Applying machine learning techniques for caching in next-generation edge networks: a comprehensive survey / J. Shuja та ін. *Journal of network and computer applications*. 2021. Т. 181. С. 103005. URL: https://doi.org/10.1016/j.jnca.2021.103005.

2. Xiao Z. Introduction of reinforcement learning (RL). *Reinforcement learning*. Singapore, 2024. С. 1–22. URL: https://doi.org/10.1007/978-981-19-4933-3_1 .

УДК 004.05

О.А. Антонюк[1,2], С.В. Науменко[1,3]
antonyk063@gmail.com, igarasym@space-it.com.ua
[1]*Черкаський національний університет імені Богдана Хмельницького, Черкаси*
[2]*Студент 4 курсу*
[3]*Науковий керівник*

ВЕБЗАСТОСУНОК ДЛЯ КООРДИНАЦІЇ СПІВПРАЦІ МІЖ ФРІЛАНСЕРАМИ ТА ЗАМОВНИКАМИ

В наш час, вебзастосунки для координації співпраці між фрілансерами та замовниками стали дуже актуальним, насамперед тому, що це дуже зручно і займає значно менше часу на пошук фахівців. Також, варто зазначити, що онлайн-платформи такого виду, зазнали популярності саме в час пандемії COVID-19. В той час, по всьому світу, відбулися локдауни, тому люди були змушені шукати альтернативні способи заробітку. Найкращими з них – стало використання таких застосунків. Онлайн-платформи дають змогу людям зі всього світу знаходити потрібні фріланс-сервіси, виконавши всього декілька кліків. Також вони гарантують безпеку виконання платежів з допомогою ескроу систем, сенс яких полягає у тому, що під час оплати, замовник перераховує кошти на спеціальний банківський рахунок онлайн-платформи, а не на рахунок виконавця на пряму. Виконавець отримує кошти, лише у разі успішного виконання роботи, або деяку суму, у разі відмови замовника від ордера.

Існує багато аналогів онлайн-платформ такого типу, серед яких найпопулярнішими є Fiverr, Upwrok та Freelancer. Fiverr пропонує фрілансерам створення власних оголошень, а клієнтам – пошук сервісів з використанням різних фільтрів та виконання замовлень. Freelancer – це онлайн-платформа, яка працює за принципом аукціону, де клієнти публікують свої завдання, а фрілансери пропонують ціну за виконання роботи. В результаті – замовник обирає фрілансера, що йому підходить. Upwork є дуже схожим на Freelancer, але на Upwork більше уваги приділяється довгостроковим контрактам, в той час як Freelancer частіше використовується для одноразових проєктів. Проаналізувавши дані аналоги, можна зробити висновок, що вебзастосунок повинен мати інтуїтивно зрозумілий інтерфейс, ефективну систему фільтрації і пошуку проєктів, механізм створення та управління замовленнями, вбудований чат та гарантії безпеки платежів.

Для реалізації проєкту, було обрано використання, розподіленої архітектури, що складається з двох частин – клієнтської та серверної. Клієнтська частина, була побудована за допомогою Svlete фреймворку. Svelte – це JavaScript фронтенд фреймворк з відкритим кодом для створення інтерактивних веб-сторінок. Загальна концепція Svelte схожа на раніше розроблені фреймворки, такі як React та Vue, оскільки вона дозволяє розробникам робити веб-додатки[1] Його перевагою є швидкодія, що зумовлено тим, що він використовує компілятор, який перетворює Svelte код в чистий JavaScript. Також в результаті цього, фінальний розмір додатку стає меншим, порівняно з аналогами. Крім цього, він має дуже простий синтаксис та підтримує реактивність за допомогою так званих «рун», тобто елементів, що виконують автоматичне оновлення інтерфейсу. Недоліком цього фреймворку є значно менша спільнота, що спричиняє меншу кількість написаних бібліотек, що може ускладнити розробку вебзастосунків.

Серверна частина була побудована на мові програмування Go. Ця мова програмування славиться своєю статичною типізацією, стабільністю, надійністю, простотою та високою продуктивністю, завдячуючи компілятору, який перетворює Go код у машинний. Фреймворком серверної частина було обрано Fiber. Fiber – це сучасний веб-фреймворк для розробки високопродуктивних веб-додатків на Go. Він створений, як один з найшвидших доступних веб-фреймворків і досягає цього завдяки Go багатопотоковості та низькорівневого контролю.[2] Його перевагами є швидкодія, завдячуючи fasthttp, простий API, який був натхненний Express.js, підтримка різних вбудованих можливостей, таких як роутинг, cookie, HTMX, WebSocket, rate-limiting, JWT та інше. Недоліком є менша спільнота, порівняно з іншими фреймворками, що може стати перешкодою при виникненні певних помилок в коді і пошуку їх рішень.

Базою даних, було обрано PostgreSQL. Перевагами даної СУБД є підтримка транзакції та ACID властивостей, розширення за допомогою плагінів та підтримка складних типів даних, таких як JSON, hstore та інше. Недоліком є відсутність вбудованої підтримки горизонтального масштабування, високі навантаження на пам'ять та центральний процесор, при роботі з великими обсягами даних.

Отже, розробка онлайн-платформи такого типу є актуальною, оскільки попит на фріланс-послуги постійно зростає. Вони забезпечують зручність та ефективність, що важливо для розвитку ринку фрілансу.

Список літератури

1. What is Svelte? URL: https://how.dev/answers/what-is-svelte (дата звернення 09.04.2025).
2. Introduction to GoLang Fiber. URL: https://withcodeexample.com/introduction-golang-fiber-web-applications (дата звернення 10.04.2025).

УДК 004.4

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

МЕТОДИ ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОСТІ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ПІДТРИМКИ ОПЕРАТОРА

Основною функцією підсистеми отримання вхідних даних інтелектуальних систем підтримки оператора (СПО) є прийом значень технологічних параметрів від зовнішнього оточення. Точками взаємодії підсистеми з зовнішнім оточенням є вхідні сигнали. Вхідні сигнали можуть бути розділені на первинні та вторинні. Значення первинних вхідних сигналів отримуються шляхом вимірювань. Прикладами таких значень є тиск, температура, тощо. Також, до первинних вхідних сигналів можна віднести інформацію, яка отримана від оператора: введені через інтерфейс користувача значення, натиснуті оператором кнопки, сформовані команди за допомогою ключів керування тощо. Вторинними сигналами можна вважати сигнали, значення яких отримані в результаті обробки даних первинних сигналів: швидкість, температура насичення, спрацювання уставок, тощо. У найпростішому випадку, вхідна інформація СПО може бути представлена у вигляді множини технологічних параметрів P:

$$P = \{P_1, P_2, P_3, \ldots, P_n\} \quad (1)$$

де P_i – окремий вхідний параметр, n – загальна кількість параметрів.

Проте таку модель не можна використовувати безпосередньо. На практиці, між параметром та алгоритмом СПО існує система вимірювання з проміжних апаратних та програмних елементів, які забезпечують прийняття, первинну обробку, перетворення, передачу інформації. Це обумовлено такими причинами: необхідність фільтрації та обробки даних у випадку наявності шумів, перешкод або нестабільності; наявність параметрів, що формуються на основі сигналів з декількох датчиків або обчислень; конверсія сигналів з одного типу в інший, стандартний та придатний для обробки; фізична віддаленість елементів не дозволяє підключити датчики безпосередньо до системи; наявність фізичних бар'єрів захисту; необхідність захисту даних від зовнішніх атак: додаткові елементи (брандмауери, системи контролю доступу, дата-діоди), що забезпечують інформаційну безпеку; необхідність протоколювання та моніторингу; координація та інтеграція великої кількості програмно-технічних комплексів (ПТК), реалізація різних протоколів зв'язку за допомогою шлюзів, тощо.

Елементи шляхів. Враховуючи наявність системи вимірювання та обробки, для кожного параметру можна визначити шляхи передачі, що складаються з множини елементів. Для представлення шляху Path використаємо векторну модель:

$$Path = \{E_1, E_2, E_3, \ldots, E_i\} \quad (2)$$

де E_{ni} – елемент, через який проходять дані сигналу.
Елементи шляху передачі параметру приведений на рисунку 1.

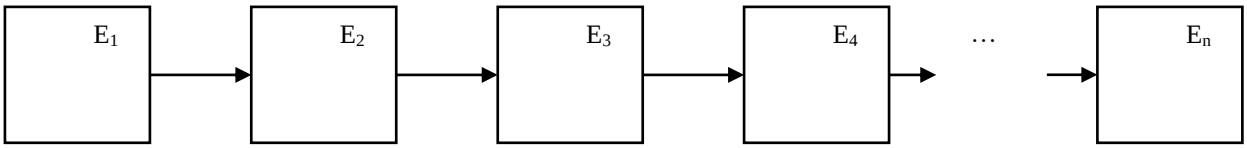


Рис. 1. Елементи шляху вхідних даних

Кожний елемент у векторі *Path* може мати набір атрибутів: тип, продуктивність, надійність, протокол обміну, час затримки, тощо. Приклад набору елементів шляху вхідних даних для параметру: E_1 – датчик; E_2 – перетворювач форми сигналу (аналоговий-цифровий); E_3 – фільтр; E_4 – кабельний канал; E_5 – комутатор; E_6 – модуль контролю безпеки; E_7 – мережева карта; E_8 – сервер; E_9 – програмне забезпечення сервера. Така модель отримання вхідних даних СПО, з точки зору теорії надійності, є найменш стійкою до відмов, оскільки передбачає послідовне з'єднання елементів.

Резервування шляхів передачі вхідних сигналів. Одним з поширених способів підвищення відмовостійкості систем вимірювання та передачі критичних параметрів є резервування. Зазвичай для вимірювання значення одного технологічного параметру застосовуються декілька датчиків, які мають

незалежну схему живлення, конструкцію та канали передачі інформації (багатоверсійний підхід). При відмові елементів одного каналу отримання даних інформація про параметр залишається актуальною через інші датчики та канали зв'язку. На АЕС можуть бути використані наступні види резервування: резервування датчиків; резервування ліній зв'язку; резервування програмного забезпечення; резервування комунікаційних пристроїв; резервування серверів тощо. Таким чином, для отримання значення одного параметра може використовуватися матриця шляхів

$$\begin{Bmatrix} Path_1 \\ Path_2 \\ \\ Path_n \end{Bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{1m} \\ E_{21} & E_{22} & E_{2m} \\ \\ E_{n1} & E_{n2} & E_{nm} \end{bmatrix}, \quad (3)$$

де E_{ij}- елементи шляхів.

При резервуванні отримується паралельна робота шляхів вимірювання (або їх частин), що підвищує відмовостійкість системи.

Слід зазначити, що створення окремих шляхів отримання вхідної інформації для кожного ПТК є складною та не завжди можливою задачею. Зазвичай інформація про критично важливі параметри отримується з резервованих датчиків, розмножується та передається у різні ПТК. В свою чергу, підсистема отримання вхідних даних СПО може отримати значення одного і того самого параметру від різних ПТК. Наприклад, інформація про температуру теплоносія в реакторі реєструється в системах захисту (АЗ), системах безпеки (СБ), системах нормальної експлуатації (СНЕ). Кожна точка реєстрації параметру в ПТК є окремим сигналом з унікальним ідентифікатором. Таким чином, СПО може отримати інформацію про значення технологічного параметру з різних сигналів, що надходять різними шляхами, з залученням різних елементів.

Багатоверсійність елементів шляхів. Кожний елемент вносить певну інтенсивність відмов у систему. Всі відмови можуть бути розподілені на дві категорії: відмови з фізичних причин та відмови через недоліки у проєктуванні.

Для оцінки інтенсивності відмов через фізичні дефекти використовуються довідкові дані виробників. Інтенсивність відмов через дефекти проєктування може бути оцінена статистичним шляхом.

Метод кількісної оцінки зменшення інтенсивності відмов при використанні двоверсійної (диверсної) системи описаний у [10].

У відповідності до даного методу, множина всіх відмов N у двоверсійній системі визначається як

$$N=N_1 \cup N_2 \cup N_{12}, \quad (4)$$

де N_1 – множина відмов тільки першої версії системи, N_2 – множина відмов тільки другої версії системи, N_{12} – множина одночасних відмов першої та другої версії системи (рис. 2).

Інтенсивність відмов двоверсійної системи визначає потужність множини N_{12}. Очевидно, що ця множина зменшується зі зростанням відмінностей у проєктуванні двоверсійної системи.

Кількісне співвідношення інтенсивності відмов двоверсійної системи та одноверсійної системи визначається коефіцієнтом диверсності:

$$K_D = \frac{|N_{12}|}{|N^{1в}|} \quad (5)$$

де $|N_{12}|$ – потужність множини відмов двоверсійної системи, а $|N^{1в}|$ – потужність множини відмов одноверсійної системи.

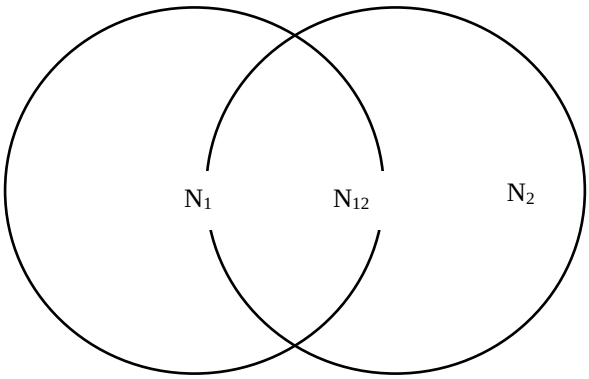


Рис. 2. Множини відмов двоверсійної системи

В результаті застосування диверсності, інтенсивність відмов двоверсійної системи λ_{12}, що викликані загальними дефектами проектування обох версій становить:

$$\lambda_{12} = K_D \cdot \lambda^{1В} \quad (6$$

)

де λ_{12} – інтенсивність відмов одноверсійної системи, K_D – коефіцієнт диверсності.

Для кількісної оцінки глибини різниці між версіями елементів системи використовуються метрики диверсності. Значення 0 метрики відповідає відсутності різниці між версіями (одноверсійна система), 4 – максимальній різниці.

Коефіцієнт диверсності K_D системи обернено пропорційний до інтегрального показника диверсності системи: при відсутності диверсності ($D = 0$) коефцієнт K_D буде рівний одиниці, що згідно (6), не знизить інтенсивність відмов з загальної причини.

Висновки. У даному дослідженні було показано, що при забезпеченні відмовостійкості системи підтримки оператора АЕС необхідно використовувати такі підходи як резервування, голосування та багатоверсійність.

Було показано, що вхідні дані до системи підтримки можуть надходити різними шляхами, що складаються з множини елементів.

Список літератури

1. 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. URL: https://doi.org/10.28925/2663-4023.2024.26.673
2. Shooman M. (2002). Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design. JOHN WILEY & SONS, INC. URL: http://dx.doi.org/10.1002/047122460X.fmatter_indsub
3. Kumar, V., Singh, L. and Tripathi, A.K. (2018), Reliability analysis of safety-critical and control systems: a state-of-the-art review. IET Softw., 12: 1-18. URL: https://doi.org/10.1049/iet-sen.2017.0053
4. Харченко В. С. Гарантоздатні системи та багатоверсійні обчислення: аспекти еволюції / В. С. Харченко // Радіоелектронні і комп'ютерні системи. 2009. №7. С.46-59. URL: http://nbuv.gov.ua/UJRN/recs_2009_7_9.
5. Kharchenko, VS, Bakhmach, ES, Siora, AA, Sklyar, VV, & Tokarev, VI. "Diversity-Oriented FPGA-Based NPP I&C Systems: Safety Assessment, Development, Implementation." *Proceedings of the 18th International Conference on Nuclear Engineering*. *18th International Conference on Nuclear Engineering: Volume 1*. Xi'an, China. May 17–21, 2010. pp. 755-764. ASME. URL: https://doi.org/10.1115/ICONE18-29754
6. Lyu, Michael & Jia-hongchen, & AlgirdasaviŽienis,. (1994). Experience in Metrics and Measurements for N-Version Programming. International Journal of Reliability, Quality and Safety Engineering. Vol| 01. No 01. pp. 41-62. URL: https://doi.org/10.1142/S0218539394000052 (lfnf pdthytyyz 11/04/2025)
7. Avizienis, Algirdas. (1995). The Methodology of N-Version Programming.
8. Harry Jones. (2012) "Common Cause Failures and Ultra Reliability," AIAA 2012-3602. *42nd International Conference on Environmental Systems*.
9. Strigini, L. & Littlewood, B. (2000). A discussion of practices for enhancing diversity in software designs. London, UK: Centre for Software Reliability, City University London. URL: https://openaccess.city.ac.uk/id/eprint/275/
10. Бахмач Є. С., Герасименко О. Д., Головір В. А., Сіора О. А., Скляр В. В., Токарєв В. І., Харченко В. С. «Стійкі до відмов інформаційно-керуючі системи на програмованій логіці». НАУ «ХАІ», НВП «Радій», Харків-Кіровоград, 2008.
11. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178. URL: https://doi.org/10.26906/SUNZ.2023.2.170
12. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.URL: https://doi.org/10.26906/SUNZ.2023.3.155.
13. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка.* 2024. №3(23), С. 111-131. URL: https://doi.org/10.28925/2663-4023.2024.23.111131
14. Вінтенко Б.Ю., Миронець І.В., Смірнов О.А. Коваленко О.В., Смірнов С.А., Буравченко К.О., Якименко Н.М. «Дослідження інформаційного забезпечення та технологічних регламентів процесів керування критичною інфраструктурою енергоблоку АЕС з реактором типу ВВЕР-1000». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка».* 2024. № 1(25), С. 253–278. URL: https://doi.org/10.28925/2663-4023.2024.25.253278

УДК 004.4

Є.І. Горбачов[1], Р.О. Ткачук[1]
zekagor0@gmail.com, tkachukroman64@gmail.com
[1]Центральноукраїнський національний технічний університет, Кропивницький

АНАЛІЗ ВИКЛИКІВ ТА ПЕРСПЕКТИВ ВИКОРИСТАННЯ ШІ В DEVSECOPS ДЛЯ ПОСИЛЕННЯ SSDLC

DevSecOps це методологія розробки ПЗ, що об'єднує Development, Security і Operations, впроваджуючи заходи безпеки на всіх етапах життєвого циклу розробки (SDLC) [1]. Тобто безпека інтегрується безперервно, а не додається як фінальний етап перед релізом. Такий підхід долає розрив між командами розробників, спеціалістів з безпеки та експлуатації, дозволяючи забезпечити захищеність конвеєрів CI/CD і випускати якісне програмне забезпечення швидко. З огляду на зростання кіберзагроз, DevSecOps перетворився з опції на необхідність сучасної розробки. SSDLC це підхід, який підсилює традиційний SDLC шляхом врахування питань безпеки на кожному етапі розробки [2]. Таким чином, SSDLC і DevSecOps мають спільну мету превентивне вбудовування безпеки в процес розробки, що підвищує стійкість ПЗ до атак.

Штучний інтелект відіграє дедалі більшу роль у DevSecOps, допомагаючи автоматизувати рутинні задачі та обробляти великі масиви даних швидше, ніж це під силу людині [3,4]. Сучасні AI/ML-технології здатні вдосконалити процеси забезпечення безпеки на різних етапах SSDLC [5,6].

Мета дослідження це сучасні підходи до інтеграції ШІ в DevSecOps-процеси та визначити їх вплив на безпеку життєвого циклу розробки ПЗ. Це охоплює огляд ролі ШІ, приклади інструментів, аналіз основних викликів впровадження та потенційні перспективи для підвищення ефективності SSDLC.

Проведені дослідження дозволили виділити наступні ключові напрями використання ШІ в DevSecOps включають:

1. Автоматизоване виявлення вразливостей. ML-алгоритми можуть швидко сканувати вихідний код, репозиторії та навіть бінарні збірки для пошуку відомих уразливостей що значно швидше і ретельніше за ручні методи. Це дозволяє виявляти проблеми на ранніх етапах розробки, пріоритезувати їх за критичністю та навіть пропонувати варіанти виправлення.

2. Покращений аналіз коду та код-рев'ю. Інструменти на основі ШІ можуть перевіряти код на відповідність найкращим практикам безпеки. Завдяки розумінню контексту і семантики коду, такі системи здатні виявляти складні уразливості, які можуть вислизнути від очей людини або не спрацьовувати на традиційні статичні аналізатори. Наприклад, GitHub Copilot (AI-асистент для програмування) допомагає розробникам писати код швидше, але водночас потребує контролю, дослідження показали, що близько 24-30% згенерованих ним фрагментів коду містять відомі уразливості.

3. Динамічне тестування та моніторинг. ШІ спрощує впровадження безперервного Security Testing. AI-орієнтовані інструменти можуть автоматично виконувати статичний і динамічний аналіз безпеки застосунків (SAST/DAST) на кожному білді, виявляючи уразливості до випуску в продакшн. Після деплойменту аналітика аномалій на основі ML відстежує поведінку користувачів та системи в режимі реального часу, щоб своєчасно сповістити про відхилення, що можуть свідчити про інцидент безпеки. Таким чином реалізується безперервний моніторинг, система автоматично вчиться нормальної поведінки і швидко реагує на підозрілі дії або аномальні навантаження.

4. Прогнозування та адаптивна безпека. Аналізуючи великі обсяги історичних даних про атаки та вразливості, ШІ здатен здійснювати прогностичну аналітику передбачати потенційні загрози і "вузькі місця" в продуктивності систем. Це дає змогу проактивно зміцнювати захист ще до того, як нові вектори атак будуть експлуатовані зловмисниками. Більш того, з розвитком adaptive security системи на базі ШІ можуть динамічно коригувати політики безпеки або оточення застосунку у відповіді на виявлені загрози. Наприклад, при фіксації нетипового трафіку AI-модуль може автоматично ізолювати підозрілий компонент або посилити фільтрацію, запобігаючи поширенню атаки.

5. Автоматизація відповідності та комплаєнсу. Інтегрований в конвеєр DevSecOps ШІ може перевіряти кожен білд на відповідність вимогам стандартів безпеки та нормативним актам (наприклад, GDPR, PCI DSS тощо). За допомогою таких інструментів компанії автоматично виконують compliance-перевірки артефактів збірки, зменшуючи людський фактор і забезпечуючи постійне дотримання необхідних стандартів.

Загалом, впровадження ШІ в DevSecOps дозволяє перейти від реактивного реагування на загрози до проактивного та навіть превентивного захисту. Машинне навчання доповнює можливості фахівців з безпеки, забезпечуючи глибший аналіз та швидкість, необхідну для захисту сучасних швидкоплинних процесів розробки .

Як приклади сучасних AI-інструментів в DevSecOps є низка комерційних і відкритих інструментів вже реалізують зазначені підходи, інтегруючи ШІ в різні етапи SSDLC. Розглянемо декілька відомих прикладів:

1. Snyk Code (DeepCode). Платформа для безпеки розробників сканує вихідний код у підтримуваних мовах (Java, JavaScript, Python, TypeScript, C/C++ тощо) і використовує кілька ML-моделей, навчених на

мільйонах зразків відкритого коду, щоб знаходити уразливості на рівні рядків і пропонувати можливі виправлення.

2. GitHub Copilot. Copilot інструмент автодоповнення коду на основі великої мовної моделі (OpenAI Codex), інтегрований в IDE. Хоча основна його мета – прискорити написання коду, побічно він впливає і на безпеку розробки. Тому Copilot доцільно застосовувати в парі з іншими засобами аналізу так як він підвищує продуктивність.

3. Amazon CodeGuru. сервіс від AWS, що використовує поєднання ML та формальних методів (automated reasoning) для автоматичного рев'ю коду та пошуку проблем продуктивності й безпекиКомпонент CodeGuru Reviewer сканує вихідний код і дає рекомендації щодо виправлення знайдених дефектів, тоді як CodeGuru Security зосереджується на пошуку вразливостей. Однією з ключових переваг є мінімізація хибних позитивів тобто глибокий семантичний аналіз коду дозволив досягти високої точності виявлення реальних вразливостей, відсіюючи більшість помилкових тривог.

4. Checkmarx One (AI Security) пропонує комплексну платформу для Application Security з можливостями, підсиленими ШІ. Зокрема, їхній статичний аналізатор (SAST) нового покоління поєднує швидкі методи сканування з ML-моделями, що забезпечує до 90% швидше сканування порівняно зі старими підходами, та до 80% менше хибнопозитивних результатів.

Еволюція ролей і процесів. З поширенням ШІ зміниться і характер роботи команд розробки та безпеки. Рутинні завдання (сканування логів, пошук відомих багів) дедалі більше автоматизуватимуться, що дозволить спеціалістам зосередитися на стратегічних аспектах – моделюванні загроз, архітектурній безпеці, реагуванні на інциденти. З'являться нові ролі, такі як ML engineer в команді безпеки, відповідальний за навчання та вдосконалення моделей, або AI-boosted аналітик безпеки. Процеси SSDLC стануть більш дано-орієнтованими: рішення про реліз можуть прийматися з урахуванням метрик ризику, згенерованих ШІ (наприклад, "рівень ризику версії" на основі знайдених уразливостей і трендів атак). Це сприятиме більш тісній інтеграції між бізнесом і IT, оскільки ризики будуть кількісно оцінюватися і зрозуміло презентуватися керівництву в реальному часі.

У результатіяк висновок штучний інтелект вже зараз переосмислює підходи до захисту ПЗ, інтегруючись у практики DevSecOps. Розв'язуючи поточні проблеми від повільного тестування до надміру сигналів, ШІ дозволяє зробити SSDLC більш швидким, безперервним та надійним. Водночас, успішне впровадження ШІ потребує усвідомлення викликів, а саме необхідно інвестувати в дані, інфраструктуру та навички команди, забезпечити прозорість і контроль рішень AI. Перспективи виглядають багатообіцяюче від автономних помічників, що самі виправляють вразливості, до повністю інтегрованих систем, де безпека розчинена в процесі розробки. Таким чином, синергія DevSecOps і ШІ здатна значно посилити безпеку життєвого циклу розробки як нині, так і в довгостроковій перспективі. Це закладає основу для нового покоління програмних продуктів, що будуть створюватися швидко, але з вбудованим захистом від сучасних загроз.

Список літератури

1. Blending AI and DevSecOps: Enhancing Security in the Development Pipeline. URL: https://devops.com/blending-ai-and-devsecops-enhancing-security-in-the-development-pipeline (дата звернення: 07.04.2025).

2. Secure SDLC. URL: https://www.wiz.io/academy/secure-sdlc (дата звернення: 07.04.2025).

3. Securing the Future: DevSecOps in the Age of Artificial Intelligence. URL: https://devops.com/securing-the-future-devsecops-in-the-age-of-artificial-intelligence (дата звернення: 07.04.2025).

4. Cheenepalli J., Hastings J. D., Ahmed K. M., Fenner C. Advancing DevSecOps in SMEs: Challenges and Best Practices for Secure CI/CD Pipelines. arXiv preprint arXiv:2503.22612. 2025. URL: https://www.arxiv.org/pdf/2503.22612

5. AI Security Tools: The Open-Source Toolkit. URL: https://www.wiz.io/academy/ai-security-tools (дата звернення: 07.04.2025).

6. AI for Cybersecurity: 8 Game-Changing Opportunities in 2024 and Beyond. URL: https://review.insignia.vc/2025/02/06/ai-cybersecurity/ (дата звернення: 07.04.2025).

УДК 004.4

К.О. Бездольний[1], А.С. Коваленко[1], О.В. Коваленко[1]
bezdolnykirill@gmail.com, annasun911@gmail.com, dr.kovalenkoov@gmail.com
[1]*Центральноукраїнський національний технічний університет, Кропивницький*

ОЦІНКА ВПЛИВУ ЛЮДСЬКОГО ЧИННИКА НА РИЗИКИ В КОМАНДНІЙ РОЗРОБЦІ ПЗ

Ризик-менеджмент є невід'ємною частиною командної розробки ПЗ (ПЗ), оскільки будь-який проєкт стикається з невизначеністю та потенційними проблемами. Особливо актуальним є врахування людського чинника, а семе сукупності характеристик і поведінки людей у команді, що впливають на результати проєкту [1,2]. Дослідження свідчать, що навіть за наявності сучасних технологій і процесів основні причини невдач IT-проєктів залишаються пов'язаними з людьми [3,4]. Іншими словами, поки в розробці залучені люди, людський фактор залишається одним із головних джерел ризику [5].

Таким чином, актуальність теми обумовлена необхідністю глибше зрозуміти, як саме людський чинник породжує ризики в командній розробці ПЗ, і як ці ризики можна передбачати та пом'якшувати.

Проведено досліджено направлено на виявлення основних проявів людського чинника, що впливають на виникнення ризиків у процесі командної розробки ПЗ. Дослідження спрямоване на ідентифікацію людських аспектів командної роботи, які найчастіше призводять до ризикованих ситуацій (затримок, дефектів, провалів проєкту тощо). Проведені дослідження показали що у командних проєктах ПЗ людський чинник проявляється у різних формах, прямо чи опосередковано породжуючи ризики. Було сформовано класифікацію людських ризиків (рис.1), до основних його проявів належать:

1. Комунікаційні проблеми.

Непорозуміння між членами команди або недостатня якість комунікації можуть призвести до неправильного тлумачення вимог і цілей проєкту. Наприклад, бізнес-вимоги можуть бути донесені не до всіх учасників або у спотвореному чи неповному вигляді, що спричиняє хибну інтерпретацію завдань розробниками.

2. Людські помилки та неуважність.

Людський фактор виявляється у різних аспектах, програмісти можуть допускати помилки через неуважність, забудькуватість або випадкове натискання клавіш, а інколи й від браку досвіду та знань. Отже, якість та надійність людської роботи безпосередньо впливає на якість продукту, а отже є джерелом ризику.

Рис. 1. Запропонована класифікація за людськими ризиками

3. Мотивація, дисципліна і вплив психологічних факторів.

Зниження мотивації або вигорання розробників можуть суттєво зменшити продуктивність праці і викликати відставання від графіка. Щоденні спокуси на кшталт відкласти на завтра чи піти раніше з роботи мають кумулятивний ефект, якщо достатньо багато учасників команди піддадуться такій поведінці, це знижує загальну трудову дисципліну і темп роботи.

4. Кадрові зміни та втрата знань.

Нестабільність команди це одна з найчастіших причин провалу проєктів. Ризик того, що проєкт раптово покине ключовий розробник, дуже непокоїть керівників ризиків. Втрата навіть одного цінного фахівця може призвести до втрати критичних знань про систему і затримати розробку на невизначений час. Особливо важко замінити незамінних співробітників, якщо знання не були задокументовані або розподілені між командою. Отже, плинність кадрів та недостатня взаємозамінність членів команди утворюють значне джерело ризику для проєкту.

5. Нереалістичні оцінки та управлінські прорахунки. Людський чинник проявляється і у фазі планування та менеджменту. Надмірний оптимізм розробників або менеджерів при оцінці термінів і трудомісткості може призвести до встановлення нереальних дедлайнів.

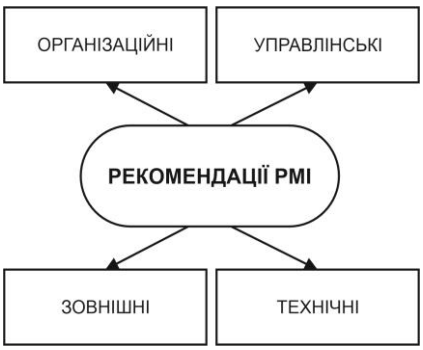


Рис. 2. Класифікація PMI за природою ризика

В результаті команда апріорі не встигає вчасно, що створює постійний ризик зриву графіка. Такі ризики часто пов'язані з недостатнім досвідом або небажанням повідомити погані новини керівництву. До управлінських проявів людського фактору також відноситься недотримання встановлених процесів (наприклад, пропуск тестування через поспіх), ігнорування ризиків у надії якось буде, недостатнє залучення клієнта чи користувачів до обговорення вимог тощо. Усі ці дії або бездіяльність мають коріння в людській природі (сподівання, припущення, комунікація) і можуть реалізувати різні ризики проєкту.

Варто зазначити, що людський чинник не діє у вакуумі, він часто посилює інші види ризиків. Наприклад, технічна складність або часті зміни вимог самі по собі є ризиками, але їх вплив значно погіршується через людські аспекти. Наприклад складне завдання легше виконати команді з високим рівнем взаємодії та досвіду, а зміни вимог менш критичні, якщо команда відкрита до спілкування з замовником і гнучкого планування. Дослідження підтверджують, що у більшості категорій ризикоутворюючих факторів простежується сильний вплив особистісного чинника.

Класифікація ризиків дозволяє впорядкувати різні загрози проєкту за їх природою. Згідно з рекомендаціями PMI (Project Management Institute), ризики в розробці ПЗ можна умовно поділити на чотири широкі категорії (рис. 2): технічні, зовнішні, організаційні, управлінські.

Людський чинник прямо або опосередковано впливає на багато з цих категорій. Найбільш очевидно він представлений в організаційних ризиках та ризиках управління проєктом, адже сюди належать команда проєкту, комунікації, розподіл ролей, мотивація тощо. Саме в цих сферах помилки чи проблеми людей здатні спричинити збої. Технічні ризики на перший погляд залежать від технологій, але й вони мають людський вимір, тобто неправильний вибір технології чи архітектури часто є результатом браку досвіду або знань (людський фактор), а критичні дефекти в коді є наслідком людських помилок. Зовнішні ризики (такі як зміна пріоритетів бізнесу замовника) менше підконтрольні команді, але й їх вплив можна пом'якшити людьми. Виділення людського фактору як окремої категорії ризиків виправдане тим, що за даними досліджень, він присутній у більшості випадків проблемних проєктів.

Як висновок можна сказати що людський чинник відіграє визначальну роль у ризик-профілі командних проєктів розробки ПЗ. Він проявляється через комунікації, компетенції, мотивацію, психологію і взаємодію людей у команді, і ці прояви часто стають першопричиною проектних проблем. Проведений аналіз показав, що основні ризики командної розробки це зриви термінів, дефекти, невдоволеність замовника, перевитрати що значною мірою генеруються або підсилюються саме людським фактором. До ключових «людських» ризик-факторів належать: неефективна комунікація, помилки і недбалість, конфлікти та втрата мотивації в команді, плинність кадрів і втрата знань, а також хиби в управлінських рішеннях. Класифікація ризиків демонструє, що людська компонента пронизує багато категорій від організаційних до технічних і тому потребує особливої уваги. Для підвищення успішності проєктів необхідно навчитися кількісно оцінювати і враховувати людський фактор.

Список літератури

1. IT Project Failure Rates: Facts and Reasons. URL: https://faethcoaching.com/it-project-failure-rates-facts-and-reasons (дата звернення: 07.04.2025).

2. 9 Risks in Software Development and How to Mitigate Them. URL: https://clockwise.software/blog/software-development-risks/ (дата звернення: 07.04.2025).

3. Huang F., Strigini L. HEDP: A Method for Early Forecasting Software Defects based on Human Error Mechanisms. arXiv preprint arXiv:2110.06758. 2021. Режим доступу: https://arxiv.org/abs/2110.06758

4. Moussaïd M., Kämmer J. E., Flandin G., Bavar N., Feufel M. A. Human Factors and their Influence on Software Development Teams - A Tertiary Study. ResearchGate. 2021. DOI: 10.1145/3474624.3474625

5. Lindstrom L., Smite D., Smite K. Agile software development projects-Unveiling the human-related success factors. Information and Software Technology. 2024. Vol. 152. P. 107045. Режим доступу: https://www.sciencedirect.com/science/article/pii/S0950584924000375

УДК 004.032.26

О.Р. Карпець, Т.О. Яровець, Є.В. Мелешко
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький, Україна

ДОСЛІДЖЕННЯ ПЕРСПЕКТИВ РОЗВИТКУ ТЕХНОЛОГІЇ DLSS ПОЗА МЕЖАМИ ІГРОВОЇ ІНДУСТРІЇ

DLSS (Deep Learning Super Sampling) є однією з найреволюційніших технологій сьогодення. У своїй основі вона використовує глибоке навчання, яке працює за допомогою тензорних ядер [2, 4] – високопродуктивних ядер, потужність яких досягається завдяки динамічній оптимізації обчислень. Вони використовуються через їх переваги, а саме, вони прискорюють навчання штучного інтелекту приблизно в 9.8 разів і збільшують швидкість математичних обчислень в 2.5 рази. Ці потужності дають змогу перетворити зображення низької якості у високодеталізовані та чіткі візуали.

Наразі цю технологію використовують лише у рамках ігрової індустрії, даючи гравцям неймовірно деталізовану графіку [1, 3].

Метою цієї роботи було проаналізувати принципи, закладені в основу досліджуваної технології, та оцінити її перспективи застосування поза межами ігрової індустрії.

Вперше NVIDIA представила DLSS разом із серією графічних карт GeForce RTX 20. Початкова версія використовувала нейронні мережі для відновлення деталей зображення, що вже в ті роки дозволяло підвищити продуктивність у підтримуваних іграх за допомогою зниження навантаження на графічний процесор завдяки тому, що воно перейшло на тензорні ядра. Це була перша версія DLSS.

Вже у 2020 році було випущено другу версію, яка принесла суттєве покращення якості зображення та продуктивності. Нова версія використовувала універсальну нейронну мережу, що дало змогу легше інтегрувати технології в різні ігри та забезпечити кращу якість зображення при вищій продуктивності.

Третя версія DLSS включала в себе функцію генерації кадрів на основі штучного інтелекту, що дозволило збільшити частоту кадрів шляхом вставки згенерованих кадрів між реальними (теперішнім та майбутнім), що значно підвищувало плавність зображення.

На виставці CES 2025 NVIDIA представила четверту версію власної технології, що включала Multi Frame Generation, дозволяючи генерувати до трьох додаткових кадрів на кожен рендерений, потенційно збільшуючи частоту кадрів до восьми разів. Також було впроваджено нові трансформерні AI-моделі для покращення Ray Reconstruction (дана технологія була впроваджена в проміжній версії –DLSS 3.5) та Super Resolution [1, 3].

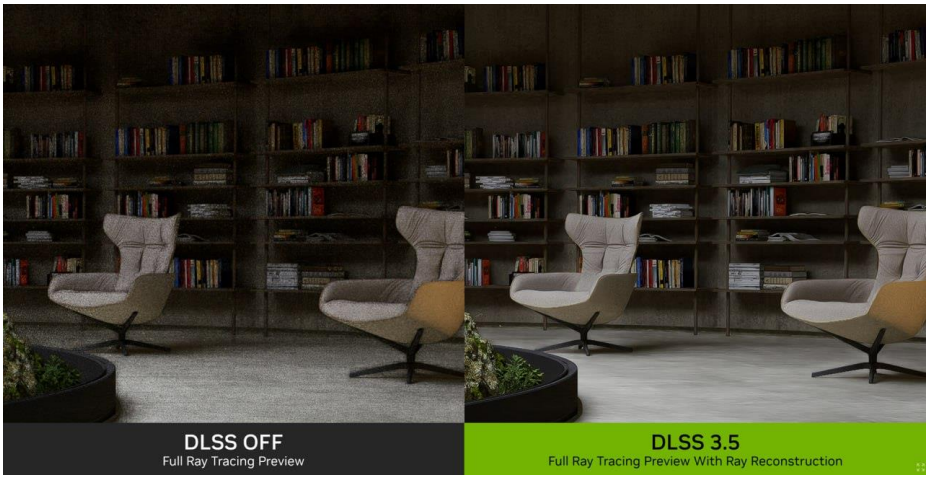


Рис. 1. Візуалізація результатів роботи NVIDIA DLSS [4]

Щоб зрозуміти потенціальні перспективи розвитку даної технології поза межами ігрової індустрії потрібно розібратися з принципами її роботи.

У традиційній графіці GPU рендерить сцену у повній роздільності монітору. Наприклад, якщо у Вас 4K монітор (3840x2160), то кожен кадр містить 8.3 мільйони пікселів. Це гігантське навантаження на систему. DLSS пропонує рендерити зображення у нижчій роздільності, наприклад, при режимі «Якість» від нативного 4K буде взято лише 66%, це врази швидше, бо графічний процесор не витрачає зайвий час та потужності на рендер пікселів, які потім все одно масштабуються. Питання якості вступає в гру: рендерити в 1080p та розтягувати до 4K – це мильно і потворно. Використання DLSS в такому випадку дає змогу «відновити» недостаючі деталі, ніби зображення початково було в 4K [2, 5].

Інтелектуальне масштабування (Super Resolution) і є тією технологією «відновлення» деталей. Основний принцип її роботи полягає у використанні нейромережі для передбачення того, як повинно виглядати зображення при високій роздільності, на основі зображення з низькою роздільністю. Мережа використовує велику кількість шарів задля виявлення різних патернів та деталей у зображеннях. Наразі найбільш детально відомо про принцип роботи трьох з них:

– Згорткових шарів, що відповідають за аналіз текстур і контурів зображення.

– Шарів активації, що допомагають мережі вирішити, яка інформація є важливою для масштабування.

– Шарів зворотного зв'язку, що використовуються для корекції помилок і покращення результатів.

При потребі, крім самого зображення, DLSS використовує різні додаткові дані для поліпшення результату:

– Motion Vectors (рух кожного пікселя по кадру, що дозволяє точніше передбачити, як повинні виглядати рухомі об'єкти).

– Depth Buffer (глибина сцени, що дозволяє правильно масштабувати об'єкти на різних відстанях).

Звісно, що «відновлення» зображення складається з певних етапів, а саме:

– *Збір тренувальних даних*. До того, як нейромережа почне працювати в реальному часі, вона проходить етап тренування на великій кількості зображень з високою роздільністю (наприклад, 16K). Створюються пари зображень: високоякісне (деталізоване зображення у 16K) та той самий кадр, зменшений до низької роздільності (наприклад, 1080p або 1440p). Даний процес дозволяє нейронній мережі вивчити, які деталі є критичними для відновлення чіткості зображення [2].

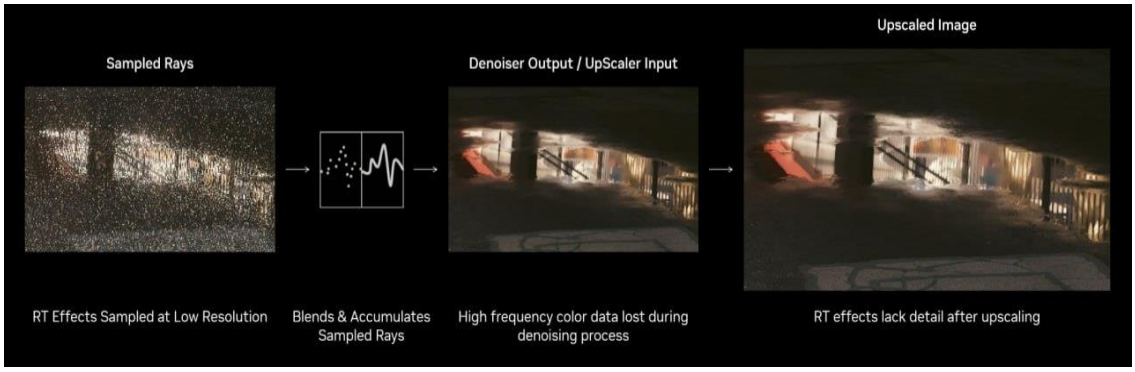


Рис. 2. Приклад вилучення шумів для подальшої роботи з зображенням [5]

– *Аналіз низькоякісного зображення*. Після того, як нейромережа була натренована, вона може працювати в реальному часі. Коли гра або додаток запускає рендеринг з низькою роздільністю (наприклад, 1080p для 4K екрану), мережа отримує це зображення на вхід та аналізує його, враховуючи фактори глибини сцени (Depth Buffer) та векторів руху (Motion Vectors).

– *Виведення високоякісного зображення*. Нейромережа виводить оброблене зображення, яке відповідає високій роздільності. Вона вміє не просто масштабувати, а й відновлювати текстури, покращувати чіткість та деталізацію, додає зображення на основі свого тренування, що включає контексти та закономірності, що були вивчені під час навчання. Тобто, замість того, щоб просто додавати пікселі за допомогою інтерполяції, нейромережа «розуміє», як виглядатиме зображення в реальній роздільності.

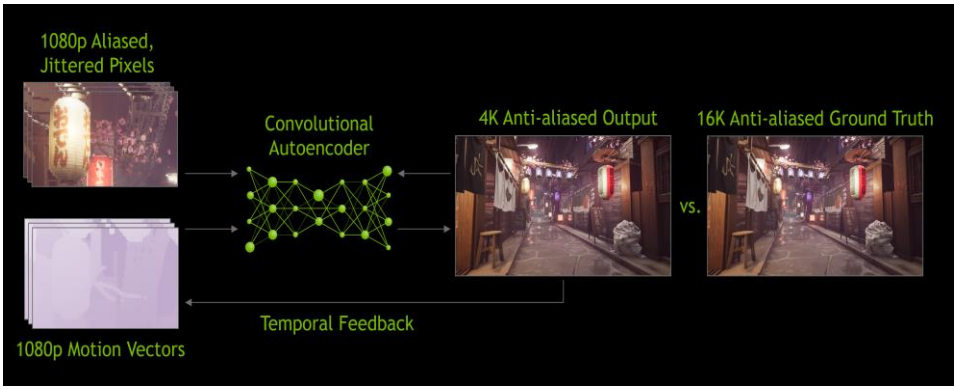


Рис. 3. Офіційна архітектура NVIDIA DLSS [2, 5]

Ми розібрали принципи, що полягають у роботі даної технології, однак її потенціал виходить далеко за

межі ігрової індустрії, особливо у критично важливих сферах, як-от медицина та військова промисловість. Використання штучного інтелекту для масштабування, реконструкції та генерації зображень може суттєво вплинути на точність діагностики, швидкість обробки даних і ефективність військових та медичних операцій.

Найважливіше, що DLSS може запропонувати сучасній медицині – це покращення візуалізації та економія часу. Сучасні методи діагностики, наприклад: МРТ, КТ, УЗД та рентгенографія, генерують величезні обсяги даних у високій роздільній здатності. Однак обробка таких зображень вимагає потужних обчислювальних систем, а час реконструкції може бути довгим, що є суттєвим фактором в деяких критичних ситуаціях, від яких може залежати життя пацієнту.

DLSS може зменшити час сканування, базуючись на тому, що система може рендерити зображення у нижчій якості, а потім відновлювати деталізацію та покращити якість знімків, використовуючи алгоритми реконструкції [5], котрі можуть відновлювати дрібні деталі (наприклад, пухлини або мікропереломи), що можуть бути нечіткими через артефакти стиснення або низьку роздільність. Також, у випадку рентгенівських досліджень можна знизити інтенсивність променів, отримуючи знімки у меншій якості, а потім відновлювати їх за допомогою нейромережі, що зменшить шкоду для пацієнтів.

У хірургії, особливо при мінімально інвазивних операціях, важливо отримувати чітке зображення в реальному часі. Виходячи з даних потреб використання DLSS дає можливості для покращення якості ендоскопічного відео (масштабування зображень у реальному часі з підвищенням чіткості без затримок).

Щодо військової промисловості, тут можна запропонувати деякі покращення, які стосуються розвідки та супутникового спостереження, наприклад, покращення якості супутникових знімків, щоб з високою деталізацією виявляти об'єкти техніки, будівель а також пересування військ. Відновлювання знімків через атмосферні перешкоди також має невід'ємний вплив на ведення дій у війні, а саме, можна запропонувати компенсування завад, таких як: дощ, туман, дим, щоб отримати чіткіше зображення. Автономних бойових систем та дронів – безпілотники та бойові роботи використовують комп'ютерний зір для навігації та ідентифікації цілей у навколишньому середовищі. Для полегшення ними розпізнавання об'єктів, DLSS може відновлювати деталі низької якості на відео, що допоможе точніше навестися на ворога. Також штучний інтелект може допомогти краще обробити відео в реальному часі, що допоможе дронам краще та швидше аналізувати місцевість й примати рішення на основі обробленого відео.

Отже, закрита та повністю монополізована технологія DLSS, що на даний час розвивається лише для ігрової індустрії, має величезний потенціал у медицині та військовій сфері. Вона може прискорити обробку зображень, покращити точність діагностики, зменшити навантаження на обладнання та підвищити ефективність розвідувальних систем. У майбутньому ми можемо побачити AI-прискорені медичні сканери, автономні бойові системи з покращеним комп'ютерним зором та супутникові системи з миттєвою обробкою даних – і все це завдяки технологіям, подібним до DLSS.

Список літератури

1. NVIDIA What is DLSS? – NVIDIA DLSS AI Technologies [Електронний ресурс]. – Режим доступу: https://developer.nvidia.com/rtx/dlss?sortBy=developer_learning_library%2Fsort%2Ffeatured%3Adesc%2Ctitle%3Aasc

2. NVIDIA DLSS 2.0: A Big Leap in AI Rendering [Електронний ресурс]. – Режим доступу: https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/

3. Digital Trends How DLSS Works [Електронний ресурс]. – Архівна версія. – Режим доступу: https://web.archive.org/web/20181102131151/https://www.digitaltrends.com/computing/everything-you-need-to-know-about-nvidias-rtx-dlss-technology/

4. Tom's Hardware What Are Tensor Cores and Their Advantages [Електронний ресурс]. – Архівна версія. – Режим доступу: https://web.archive.org/web/20200621195631/https://www.tomshardware.com/news/nvidia-tensor-core-tesla-v100,34384.html

5. NVIDIA DLSS 3.5 Ray Reconstruction [Електронний ресурс]. – Режим доступу: https://www.nvidia.com/en-ph/geforce/news/nvidia-dlss-3-5-ray-reconstruction/

УДК 004.4

Я.О. Козлов[1], А.С. Коваленко[1], О.В. Коваленко[1]
kozlov.yan1@gmail.com, annasun911@gmail.com, dr.kovalenkoov@gmail.com
[1]Центральноукраїнський національний технічний університет, Кропивницький

ЗАСТОСУВАННЯ FMEA ДЛЯ КІЛЬКІСНОЇ ОЦІНКИ РИЗИКІВ У РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Failure Mode and Effects Analysis (FMEA, аналіз видів і наслідків відмов) це методологія управління ризиками, що систематично виявляє потенційні режими відмов (причини та способи, якими система чи процес може зазнати невдачі) та оцінює їхній вплив. У контексті розробки програмного забезпечення (ПЗ) FMEA дозволяє на ранніх етапах прогнозувати проблеми та запобігати їм до того, як вони вплинуть на проект або продукт. Ключовим інструментом FMEA є показник пріоритетності ризику (RPN, Risk Priority Number) добуток оцінок серйозності наслідків, ймовірності появи та виявлюваності проблеми. RPN надає кількісну міру ризику, дозволяючи ранжувати потенційні проблеми за пріоритетом усунення. Такий підхід широко використовується у різних галузях (від NASA до промисловості) завдяки простоті і ефективності [1]. Водночас дослідники відзначають і обмеження класичного RPN наприклад, можливі неоднозначності у пріоритизації ризиків та залежність оцінок від експертних суджень [2].

Мета дослідження виявили ефективність застосування FMEA у існуючих різних моделях життєвого циклу розробки ПЗ, а саме каскадної (Waterfall), гнучкої (Agile), V-моделі, спіральної моделі з акцентом на ефективність кількісної оцінки ризиків у кожній з них.

Як приклад доказу на підтримку доцільності використання FMEA це досвід галузі, який демонструє ефективність FMEA в різних умовах. Зокрема, дослідження для ERP-систем показало, що використання FMEA протягом життєвого циклу впровадження ERP дозволило виділити топ-5 ризиків за величиною RPN (наприклад, брак залученості керівництва, зміни у масштабі проекту тощо) і виробити рекомендації для проактивного захисту від цих проблем [3]. В сфері медичних пристроїв компанії відзначають, що FMEA навіть при переході до Agile забезпечує стабільність і безпеку розробки на ранніх стадіях, що виграє всім учасникам від виробника до кінцевого пацієнта [4]. У автомобільній індустрії поєднання V-моделі та FMEA є настільки природним, що стандарти ASPICE і ISO 26262 фактично інтегрують ці підходи FMEA виступає дизайн-інструментом для усунення відмов на рівні системної та компонентної розробки [5]. Навіть у розвитку програмного забезпечення загального призначення спостерігається тренд до ширшого застосування FMEA ризик-орієнтоване тестування і DevOps-практики звертаються до аналізу відмов для підвищення надійності продуктів [6]. Отже, незалежно від того, чи йде мова про лінійну каскадну розробку, чи про гнучкі ітерації, аналіз видів і наслідків відмов залишається ефективним засобом кількісної оцінки і пріоритизації ризиків, потрібно лише правильно «вбудувати» його в обрану модель процесу. Це забезпечує прийняття технічно обґрунтованих рішень і сприяє успішному завершенню проєктів із мінімальними невдачами.

Waterfall (каскадна модель). FMEA добре вписується в послідовну структуру, дозволяючи проаналізувати ризики на стадії планування і дизайну. Це підвищує надійність класичних проектів, де зміни дорого коштують. У каскадній моделі FMEA доцільна для програмних проектів з високими вимогами до якості (наприклад у оборонній чи медичній галузі) або де наслідки збоїв критичні. Переваги застосування FMEA у Waterfall це превентивність і повнота аналізу. Недоліки застосування FMEA у Waterfall це те що важко врахувати все на початку та адаптуватися до змін. Тому для Waterfall варто застосовувати FMEA у помірному обсязі, зосереджуючись на найбільш критичних підсистемах і на вимогах, де невдача неприпустима.

Гнучка розробка програмного забезпечення (Agile software development). Agile це гнучкі методи традиційно менше формалізують управління ризиками, але практика показує вигоду від вбудовування FMEA в Agile-процеси. Адаптована Agile FMEA це регулярний, циклічний аналіз найважливіших ризиків кожного спринту чи релізу. Переваги застосування FMEA у Agile. Такий підхід рекомендований для команд, що працюють над довготривалими продуктами, де накопичується технічний борг, або у критичних доменах, що переходять на Agile (наприклад в медичних продуктах, де якість вирішальна). Недоліки застосування FMEA у Agile. FMEA потрібно спростити і прискорити, щоб вона не суперечила принципам Agile, фокус тільки на кілька найкритичніших ризиків, інтеграція в зустрічі команди, мінімальна бюрократія. Доцільність FMEA в Agile зростає зі складністю системи та вимогами до надійності для простих веб-додатків, можливо, це буде зайвим, але для технічно складних проектів, навіть розробленого Agile-підходом, FMEA надає додатковий рівень впевненості у якості. Компроміс це робити FMEA менш формально, але часто, і забезпечити підтримку з боку всіх учасників процесу.

V-модель. Це середовище, де FMEA фактично стала стандартом де-факто. V-модель використовується там, де неприпустимі збої, отже аналіз відмов обов'язковий елемент процесу (в деяких сферах прямо обумовлений нормативами). Переваги застосування FMEA у V-моделі. Тут найповніше розкривається, її результати ведуть до внесення змін у дизайн, додавання резервних механізмів, розробки тестів на відмови і т.д. Без FMEA важко уявити собі успішну реалізацію, скажімо, системи управління автомобільними гальмами за V-

моделлю ризики занадто великі. Недоліки застосування FMEA у V-моделі. Водночас, обмеження RPN і громіздкість процесу вимагають постійного удосконалення методик. Рекомендується використовувати вдосконалені підходи FMEA (наприклад, введення вагових коефіцієнтів, методи типу Fuzzy-FMEA) для подолання недоліків класичного RPN, особливо коли в проекті сотні ризиків. В цілому, FMEA у V-моделі абсолютно доречна і повинна адаптуватися під стандарти доменної області. При правильній реалізації вона значно знижує імовірність відмов продукту і сприяє успішній сертифікації.

Спіральна модель. Spiral зі своєю гнучкістю та орієнтацією на ризик природньо гармонує з FMEA. Використання FMEA в кожному циклі спіралі додає суворості і кількісної основи ризик-орієнтованому керуванню. Це особливо корисно для інноваційних R&D проектів, де потрібно приймати рішення в умовах невизначеності. FMEA структурно підказує, що досліджувати та прототипувати насамперед (виходячи з RPN). Spiral-модель за допомогою FMEA може забезпечити, що команда не упустить жодного великого ризику і послідовно зменшує їх від витка до витка. Однак, для успішності потрібно, щоб команда була достатньо зрілою в аналізі ризиків; якщо бракує експертизи, FMEA на ранніх етапах може бути умовною. Загалом, FMEA доцільно застосовувати в спіральних проектах високої складності або вартості, де провал недопустимий, вона надасть необхідну видимість ризиків для ухвалення рішень "продовжувати/змінювати курс/зупинити проект". У менш критичних спіральних проектах (напр., дослідницьких невеликого масштабу) можна обмежитися неформальним аналізом ризиків, оскільки формальна FMEA може бути зайвою перепоною.

Як результат досліджень можна сказати що FMEA є універсальним інструментом кількісної оцінки ризиків, але ступінь її користі та форма реалізації залежать від моделі життєвого циклу. Планові моделі (Waterfall, V) надають сприятливе середовище для ґрунтовного одноразового аналізу тут FMEA рекомендується для забезпечення високої якості і надійності, особливо в критичних галузях. Гнучкі підходи (Agile, Spiral) виграють від FMEA, якщо адаптувати її у безперервний процес, вона підсилює проактивне управління ризиками без втрати гнучкості, хоча й потребує культури та дисципліни в команді.

В кожному випадку FMEA повинна бути масштабована під потреби проекту від легкого якісного аналізу кількох ризиків до глибокої кількісної моделі з сотнями сценаріїв. Практика показує, що ефективність FMEA висока, коли її результати безпосередньо впливають на рішення планування спринтів, вибір архітектури, введення тестових сценаріїв, тощо. Якщо ж FMEA робиться формально і не інтегрується у процес, її користь мінімальна. Отже, доцільність застосування FMEA визначається не лише моделлю розробки, а й критичністю проекту для високоризикових проектів варто впроваджувати FMEA або її елементи незалежно від методології, адаптуючи формат (спринтовий, фазовий чи циклічний) до конкретного процесу. В підсумку, вміння правильно застосувати FMEA у відповідній фазі життєвого циклу підвищує шанси успішної реалізації програмного продукту з мінімумом неприємних сюрпризів на шляху.

Список літератури

1. Zalэskan, Э. P., & Paksoy, T. K. (2024). Risk Prioritizing with Weighted Failure Mode and Effects Analysis in Software Testing Processes. Applied Sciences, 14(24), 11573. DOI:10.3390/app142411573.

2. V. Anes, T. Morgado, A. Abreu, J. Calado, L. Reis. Updating the FMEA Approach with Mitigation Assessment Capabilities - A Case Study of Aircraft Maintenance Repairs. Fracture & Failure Prevent: Reliability, Proactivity and Practice 2022. DOI:10.3390/app122211407

3. H. Shirouyehzad, D. Reza, M. Badakhshian The FMEA Approach to Identification of Critical Failure Factors in ERP Implementation. International Business Research Vol. 4, No. 3; July 2011 DOI: 10.5539/ibr.v4n3p254

4. Risk management in times of agile change. URL: https://www.dietz-consultants.com/en/pressemitteilungen/risikomanagement-in-zeiten-agiler-veraenderung-28-05-18 (дата звернення: 07.04.2025).

5. ISO 26262 Functional Safety article -5, FMEA (Failure Mode and Effects Analysis). URL: https://www.linkedin.com/pulse/iso-26262-functional-safety-article-5-fmea-failure-mode-sabari-rajan-ifo7c (дата звернення: 07.04.2025).

6. Kar Neelov Customizing system testing discipline for agile. Paper presented at PMI Global Congress 2012 North America, Vancouver, British Columbia, Canada. Newtown Square, PA: Project Management Institute. URL: https://www.pmi.org/learning/library/customizing-system-testing-discipline-agile-5975

УДК 004.75

Є.В. Мелешко, М.С. Якименко, В.В. Міхав, Я.П. Шуліка
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький, Україна

МАТЕМАТИЧНА МОДЕЛЬ ВИЯВЛЕННЯ АНОМАЛЬНИХ ЗВ'ЯЗКІВ МІЖ КОМПОНЕНТАМИ СКЛАДНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ НА ОСНОВІ ПОЛОЖЕНЬ ТЕОРІЇ ДИНАМІЧНОГО ХАОСУ

Високонавантажені веб-сервіси та складні комп'ютерні системи та мережі становлять ключову частину сучасної інфраструктури, оскільки забезпечують стабільну роботу банківських послуг, комерційних платформ, онлайн-освітніх ресурсів, соціальних мереж та багатьох інших критичних галузей. З огляду на значне навантаження, яке ці системи щоденно витримують, для їхньої стабільної роботи необхідно дотримуватися високих вимог до показників безпеки та надійності. Одна з головних проблем для таких систем – це виявлення аномалій у реальному часі. Аномалії можуть сигналізувати про несправності у функціонуванні системи, невідповідності в процесах або потенційні кібератаки. Високонавантажені системи є особливо чутливими до навіть незначних збоїв, оскільки вони можуть спричиняти значні затримки або повну недоступність сервісу для багатьох користувачів одночасно, що призводить до фінансових втрат та втрати довіри клієнтів. Сучасні веб-сервіси стикаються з такими проблемами як спроби DDoS-атак, значні перепади в запитах користувачів, проблеми з підключенням до баз даних, витоки пам'яті, а також вплив несподіваних змін у конфігурації мережі чи обладнання. Тому забезпечення своєчасного виявлення аномалій є критично важливим аспектом безпеки високонавантажених систем. Автоматизація процесу аналізу та своєчасна ідентифікація потенційних проблем в режимі реального часу дозволяють зменшити ризики та мінімізувати час простою.

Метою цієї роботи є розробка математичної моделі виявлення аномальних зв'язків між компонентами складної комп'ютерної системи в умовах обмежених ресурсів на основі положень теорії динамічного хаосу. Це дозволить підвищити швидкість виявлення аномалій у поведінці високонавантажених складних комп'ютерних систем що, в свою чергу, повинно підвищити її безпеку.

Розглянуто динаміку поведінки складної комп'ютерної системи, представлену в тривимірному просторі. Оцінимо три основні характеристики – завантаження пам'яті, процесора та мережевого пристрою. Для цієї оцінки застосуємо основні положення теорії динамічного хаосу [1, 2].

Розглянуто такий випадок, у якому інваріантним атрактором системи є граничний цикл. Цей цикл може задаватися деяким ϕ-періодичним розв'язком $x=o(t)$, де $x_0=o(0)$ є фіксованою точкою циклу. Розв'язок на інтервалі $[0, \phi)$ задає природну параметризацію точок циклу: $M=\{o(t) \mid 0 \le t \le \phi\}$. Зроблено припущення, що цикл атрактору M є E-стійким. Тоді навколо цього циклу формується стаціонарно розподілений набір випадкових траєкторій складної системи. В такому разі визначення стохастичної чутливості атрактору M відповідно до теорії динамічного хаосу для складної комп'ютерної системи зводиться до побудови та аналізу ф-періодичного розв'язку $W(t)$ матричного виразу:

$$V = F(t)V + VF^T(t) + P(t)S(t)P(t), \tag{1}$$

з T-періодичними коефіцієнтами:

$$\mathrm{F}(\mathrm{t}) = \partial f / \partial x(\xi(t)), \;\; S(t) = G(t)G^T(t),$$
$$G(t) = \sigma(\xi(t)), \;\; P(t) = P_{\xi(t)}.$$

Матриця $W(t)$, що є функцією стохастичної чутливості циклу M, являється єдиним розв'язком рівняння (1) у просторі U симетричних матриць nЧn, які визначені і є достатньо гладкими на R^1 з наступними умовами періодичності:

$$\forall t \in R^1 : V(t+\tau) = V(t), \tag{2}$$

та вродженості:

$$\forall t \in R^1 : V(t)r(t) = 0, r(t) = f(\xi(t)). \tag{3}$$

Для ймовірнісної інтерпретації матриці $W(t)$ розглянемо наступну стохастичну систему:

$$dy = F(t)ydt + P(t)G(t)dw(t). \tag{4}$$

Ця система має періодичний режим, який пов'язаний із розв'язком $\bar{y}(t)$. Отже, шуканим розв'язком $W(t)$ системи (1)–(4) і є коваріаційна матриця випадкового періодичного процесу $\bar{y}(t)$, що відповідає стабільному стану складної системи. Вона є результатом асимптотичної поведінки процесу і забезпечує характеристику зв'язків між компонентами системи у довгостроковій перспективі.

Випадкові траєкторії лінійної складної системи створюють навколо циклу набір, що лежить у деякому інваріантному околі системи в області U. Нехай PL_t – це гіперплощина, яка ортогональна циклу в точці $o(t)$

($0 \le t \le Q$). Тоді позначимо через U_t окіл точки $o(t)$, що знаходиться в PL_T: $U_t = U \cap PL_t$. В такому разі можна зробити припущення, що $U_t \cap U_s = \emptyset$, якщо $t \ne s$.

Ймовірнісний опис набору цих випадкових траєкторій зручно пов'язати з векторною функцією X_t. Значення X_t є точками перетину випадкових траєкторій лінійної складної системи в області U_t. Ймовірнісний розподіл набору траєкторій з часом стабілізується, тому випадкова змінна X_t в околі оюласті U_t має певний стаціонарний розподіл з деякою густиною $c_t(x, e)$.

Було використано геометричне представлення для дослідження розкиду випадкових траєкторій навколо циклу. На рис. 1 у вигляді зірочок наведено точки перетину випадкових траєкторій складної системи із січною площиною PL_t, ортогональною циклу у точці $o(t)$. Коваріація розподілу цих точок задана матрицею $W(t)$, яка є єдиним розв'язком вище наведеної системи рівнянь (1)–(4).

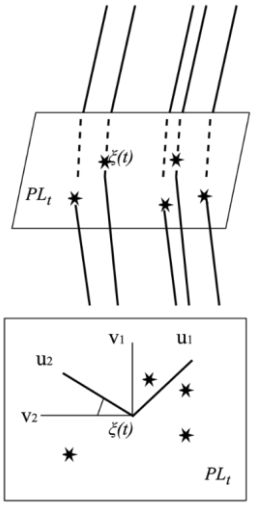


Рис. 1. Точки перетину випадкових траєкторій складної системи із гіперплощиною PL_t, ортогональною до циклу в точці $\xi(t)$

У розглянутому тривимірному просторі (n=3) для побудови рішення $V(t)$ рівняння (1) будемо використовувати наступний сингулярний розклад:

$$V(t) = \lambda_1(t) v_1(t) v_1^T(t) + \lambda_2(t) v_2(t) v_2^T(t) + \lambda_3(t) v_3(t) v_3^T(t),$$

де $л_1(t) \ge л_2(t) \ge л_3(t)$ – власні значення, а $v_1(t)$, $v_2(t)$, $v_3(t)$ – власні вектори матриці $V(t)$. З умови (3) виходить, що для будь-якого t матриця $V(t)$ є виродженою і розподіл точок перетину випадкових траєкторій складної системи (зірки на рис. 1) зосереджений на площині PL_t. А це, в свою чергу, означає, що $л_3(t)=0$ і відповідний власний вектор $v_3(t)=r(t)/\|r(t)\|$ є дотичними до циклу. Тому розклад матриці $V(t)$ має наступний вигляд:

$$V(t) = \lambda_1(t) v_1(t) v_1^T(t) + \lambda_2(t) v_2(t) v_2^T(t). \quad (5)$$

Тут $V(t)$ задається векторами $v_1(t)$, $v_2(t)$ та скалярними функціями $л_1(t)$, $л_2(t)$. Функції $л_1(t)$, $л_2(t)$ у випадку невироджених шумів строго позитивні і визначають при будь-якому t дисперсію випадкових траєкторій циклу вздовж векторів $v_1(t)$, $v_2(t)$. Значення $л_1(t)$, $л_2(t)$ задають розмір, а $v_1(t)$, $v_2(t)$ – напрямок осей еліпса розсіювання точок перетину випадкових траєкторій складної системи з площиною PL_t. Рівняння цього еліпса у площині PL_t має вигляд:

$$\left(x - \xi(t)\right)^T W^+(t) \left(x - \xi(t)\right) = 2k^2,$$

де k – довірча ймовірність $P=1-e^{-k}$.

Позначимо через $u_1(t)$, $u_2(t)$ ортонормований базис площини PL_t. Цей базис може бути знайдений за відомим τ-періодичним розв'язком $o(t)$. Власні вектори $v_1(t)$,$v_2(t)$ можна отримати обертанням базису $u_1(t)$, $u_2(t)$ на деякий кут $\phi(t)$:

$$v_1(t) = u_1(t) cos\phi(t) + u_2(t) sin\phi(t), \quad (6)$$

$$v_2(t) = -u_1(t) sin\phi(t) + u_2(t) cos\phi(t). \quad (7)$$

В результаті рівняння (5)–(7) дозволяють виразити невідоме рішення системи (1)–(3) через три скалярні функції $л_1(t)$, $л_2(t)$, $\phi(t)$.

Позначимо:

$$P_1(t) = v_1(t) v_1^T(t),\ P_2(t) = v_2(t) v_2^T(t).$$

Відзначено, що $P_i(t)$ (i=1, 2) є проекційними матрицями:

$$P_i v_i = v_i,\ P_i v_i = 0 (i \ne j),\ P = P_1 + P_2. \quad (8)$$

Представлено розклад (5) у вигляді:

$$V(t)=\lambda_1(t)P_1(t)+\lambda_2(t)P_2(t).$$

Зроблено наступне припущення. Для ортонормованих векторних функцій $v_i(t)$ і проекційних матриць $P_i(t)=v_i(t)v_i^T(t)$ (i=1,2) справедливі наступні тотожності:

$$v_1^T(t)P_1(t)v_1(t)\equiv 0, \quad (9)$$

$$v_1^T(t)P_2(t)v_1(t)\equiv 0, \quad (10)$$

$$v_2^T(t)P_1(t)v_2(t)\equiv 0, \quad (11)$$

$$v_2^T(t)P_2(t)v_2(t)\equiv 0, \quad (12)$$

$$v_1^T(t)P_1(t)v_2(t)=\hat{\varphi}(t)+\hat{u}_1^T(t)u_2(t). \quad (13)$$

$$v_1^T(t)P_2(t)v_2(t)=\hat{\varphi}(t)+\hat{u}_1^T(t)u_2(t). \quad (14)$$

Довести вказане припущення можна таким чином – тотожність (9) безпосередньо випливає з наступного співвідношення:

$$v_1^T P_1 v_2 = v_1^T\left[v_1\hat{v}_1^T\right]v_1 = v_1^T\left[v_1\hat{v}_1^T+v_1v_1^T\right]v_1 = v_1\hat{v}_1^T+v_1v_1^T=\left[v_1v_1^{\hat{T}}\right]\equiv 0. \quad (15)$$

Тотожність (12) доводиться аналогічно. Тотожність (10) отримується з:

$$v_1^T P_2 v_2, = v_1^T\left[v_2\hat{v}_2^T+v_2v_2^T\right]v_1 = v_2v_1^Tv_1\hat{v}_2^T+\hat{v}_2v_1^Tv_1v_2^T\equiv 0. \quad (16)$$

Тотожність (11) доводиться аналогічно.

Використаємо наступні рівності:

$$\hat{v}_1=\hat{u}_1cos\varphi+\hat{u}_2sin\varphi+v_2\hat{\varphi},$$

$$\hat{u}_1^Tu_1\equiv 0,\hat{u}_2^Tu_2\equiv 0,\ -\left(\hat{u}_1^Tu_2\right)=u_1^Tu_2.$$

Після цього можна сформувати такий вираз:

$$v_1^T P_1 v_2 = v_1^T\left[v_1\hat{v}_1^T+v_1v_1^T\right]v_2 = v_2\hat{v}_1^T=\left(\hat{u}_1^Tcos\varphi+\hat{u}_2^Tsin\varphi+v_2^T\hat{\varphi}\right)v_2=\left(\hat{u}_1^Tcos\varphi+\hat{u}_2^Tsin\varphi\right)v_2+\hat{\varphi}=$$

$$=-\left(\hat{u}_1^Tu_1cos\varphi sin\varphi+\hat{u}_1^Tu_2cos^2\varphi-\hat{u}_2^Tu_1sin^2\varphi+\hat{u}_2^Tu_2cos\varphi sin\varphi+\hat{\varphi}=\hat{u}_1^Tu_2+\hat{\varphi}\right).$$

З цих співвідношень слідує (13). Тотожність (14) доводиться аналогічно.

Доведене припущення надає набір тотожностей для ортонормованих векторних функцій і проекційних матриць. Ці тотожності дозволяють суттєво спростити обчислення, пов'язані з аналізом чутливості циклів і взаємодій між проекціями векторів у складній ком'ютерні системі. Оскільки матриці $P_i(t)$ є проекційними та ортогональними, їх використання допомагає розділяти аналіз векторів у різних напрямках, що робить можливим більш ефективне моделювання та прогнозування станів складної комп'ютерної системи.

Використання проекційних матриць і ортогональних векторів для вивчення поведінки компонентів складної комп'ютерної системи є відмінним підходом від існуючих моделей аналізу аномалій, таких як кореляційний аналіз або аналіз на основі кластеризації. Модель (9)–(16) надає можливість оцінювати зміни в ортогональності векторів, що дозволяє виявляти складні взаємодії між компонентами.

В результаті дослідження розроблено математичну модель виявлення аномальних зв'язків між компонентами складної комп'ютерної системи, яка на відміну від інших використовує геометричний підхід, де аномалії виявляються через зміну взаємної ортогональності між компонентами. Це дозволило зменшити час виявлення аномалій стану комп'ютерної системи до 10%. При цьому точність виявлення аномалій залишилась на заданому рівні.

Проведено дослідження використання розробленої моделі у комплексі з моделями Isolation Forest, Autoencoder, One-Class SVM. Результати досліджень показали суттєві (до 10 разів) збільшення швидкості виявлення аномалій поведінки комп'ютерної системи, при незначному зниженні точності цієї операції. Це дає можливість зробити висновки про доцільність використання розробленої моделі для виявлення аномалій поведінки високонавантажених складних комп'ютерних систем в режимі реального часу.

Список літератури

1. Devaney, Robert. (2021). An Introduction to Chaotic Dynamical Systems. DOI: 10.1201/9780429280801.
2. Göcs, László & Johanyák, Zsolt. (2023). Identifying Relevant Features of CSE-CIC-IDS2018 Dataset for the Development of an Intrusion Detection System. DOI: 10.48550/arXiv.2307.11544.

УДК 004.42:004.89

Л.В. Константинова *викл.,* А.Я. Клюй *ст. 2-го курсу*
liliyashel1976@gmail.com, nastya.klui@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

ОГЛЯД ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВЕБДИЗАЙНЕРІВ

Штучний інтелект (ШІ) дедалі активніше впроваджується у вебдизайн, надаючи інструменти для автоматизації рутинних процесів, покращення якості роботи та оптимізації витрат. Водночас велика кількість доступних рішень ставить перед фахівцями питання про їхню реальну ефективність і доцільність застосування. Тому необхідно детально розглянути сучасні ШІ-засоби, оцінити їхній функціонал та вплив на професійну діяльність дизайнерів.

Мета дослідження - оглянути та систематизувати ШІ-інструменти, які допомагають вебдизайнерам у генерації графіки, розробці UX/UI текстовій підтримці та UX аналітиці.

У роботі розглянуто вісім ШІ-рішень, які найчастіше згадуються у спеціалізованих публікаціях і форумах вебдизайнерів [1-5]. Основними критеріями відбору стали:

1. Рівень популярності (частота згадувань у професійних спільнотах та на галузевих конференціях).
2. Різноманіття функцій (генерація візуального контенту, автоматизація дизайну, UX-аналітика, текстова підтримка тощо).
3. Доступність для широкого кола користувачів (наявність безкоштовних планів або демоверсій).
4. Орієнтованість на актуальні потреби ринку (швидкість та зручність інтеграції в робочий процес).

Більшість з розглянутих досліджень зосереджені на графічних або текстових рішеннях. В даній роботі пропонується комплексний підхід, що охоплює весь процес вебдизайну – від швидкого прототипування (Uizard) та UX-аналітики (Hostinger AI Heatmap) до точкових завдань (Remove.bg) і генерації різноманітного контенту (DALL-E AI, MidJourney, ChatGPT). Такий огляд допоможе вебдизайнерам вибрати оптимальний набір сервісів та адаптувати їх до різних етапів розробки залежно від конкретних потреб.

На основі детального аналізу джерел [1-5] було сформовано таблицю з оглядом засобів ШІ (табл. 1), у якій відображено основні аспекти:

- Призначення (тип завдань, які вирішує інструмент);
- Основні можливості (які завдання допомагає вирішувати);
- Переваги (що робить інструмент привабливим у використанні);
- Недоліки (обмеження безкоштовної версії, складність налаштувань тощо);
- Ціна/доступність (наявність безкоштовного плану, вартість підписок).

Розширений опис у контексті (табл. 1) свідчить про значний вплив штучного інтелекту на сферу веб-дизайну. Розглянуті інструменти демонструють широкий спектр можливостей – від генерації візуального контенту (DALL-E AI, MidJourney) до автоматизації створення UX/UI-дизайну (Uizard) та аналізу поведінки користувачів (Hostinger AI Heatmap). Окремо варто відзначити розвиток текстових ШІ-рішень, які спрощують наповнення вебсайтів якісним контентом (ChatGPT). Всі інструменти мають як суттєві переваги, так і певні обмеження, що впливає на їхню ефективність у різних аспектах веб-дизайну.

Аналіз дослідження (табл. 1) підтверджує, що найбільшу користь штучний інтелект приносить у чотирьох основних напрямках:

1. **Генерація графічного контенту** (DALL-E AI, MidJourney, Remove.bg) – суттєво скорочує час, який витрачається на пошук або створення ілюстрацій.
2. **Автоматизація дизайну** (Uizard, 10Web AI Website Builder, Adobe Sensei) – прискорює процес розробки інтерфейсів і вебсайтів.
3. **Текстова підтримка** (ChatGPT) – допомагає оперативно наповнити сайт контентом, редагувати чи локалізувати тексти.
4. **UX-аналітика** (Hostinger AI Heatmap) – дозволяє вебдизайнерам краще адаптувати сайти під поведінку користувачів.

Проте, як показано в таблиці, кожен інструмент має свої обмеження, зокрема, точність роботи, складність налаштувань, доступність безкоштовних версій тощо. Наприклад, MidJourney та Remove.bg надають обмежену кількість запитів у безкоштовній версії, що може бути критично для великих проєктів. Генеративні системи (DALL-E AI, ChatGPT) можуть видавати нерелевантні результати без чітких інструкцій. Деякі сервіси не мають повної україномовної підтримки, що ускладнює роботу локальних дизайнерів. Adobe Sensei вимагає підписки на Creative Cloud, а 10Web AI Website Builder – на власну платформу. Для окремих дизайнерів це може бути фінансово невигідно. Це означає, що вибір ШІ-засобів у вебдизайні має бути виваженим та залежати від конкретних завдань.

Таблиця 1

Огляд засобів ШІ для вебдизайнерів

Назва	Призначення	Основні можливості	Переваги	Недоліки	Ціна/доступ-ність
DALL-E AI	Генерація зображень за текстовим запитом	Генерація високоякісних зображень у різних стилях	Висока якість, підтримка різних художніх стилів	Потребує точних запитів для якісного результату	Частково безкоштовний (50 запитів/міс.)
MidJourney	Створення художніх AI-ілюстрацій	Створення складних художніх ілюстрацій	Глибокий рівень дета-лізації, креа-тивний підхід	Не завжди правильно передає деталі	Платний (від $10/міс.)
ChatGPT	Автоматична генерація текстового контенту	Генерація текстів для веб-контенту, блогів, маркетингу.	Швидке створення текстів для різних потреб	Може давати неточні або невідповідні відповіді	Безкоштовний / Pro ($20/міс.).
Uizard	Швидке створення UX/UI макетів	Автоматизація процесу створення інтерфейсів	Простий інтерфейс, автоматизація UX-дизайну	Обмежені можливості для складного UI-дизайну	Платний (від $12/міс.)
Hostinger AI Heatmap	Аналіз поведінки користувачів на веб-сайтах	Візуалізація теплових карт, UX-аналітика	Оптимізація розташування елементів на сайті	Доступний лише у рамках платформи Hostinger	Включено у тарифні плани Hostinger
Remove.bg	Автоматичне видалення фону з фото	Миттєве видалення фону зображень	Економія часу на редагування фото	Може не коректно визначати межі об'єкта	Безкоштовний / Платний
10Web AI Website Builder	Генерація веб-сайтів за допомогою ШІ	Швидке створення сайтів без програмування	Зручність для новачків у веб-розробці	Менше можливостей для гнучкого редагування.	Платний (від $10/міс.)
Adobe Sensei	AI-автоматизація дизайну в продуктах Adobe	Автоматична оптимізація зображень, розпізнавання	Глибока інтеграція з продуктами Adobe	Доступний тільки у рамках Adobe Creative Cloud	Доступний у Adobe Creative Cloud

Висновки. Проведений огляд підтверджує, що штучний інтелект здатен суттєво підвищити ефективність вебдизайнерів, але вибір конкретного інструмента залежить від обсягу та специфіки проєкту, а також доступного бюджету. Завдяки комплексному підходу до порівняння було виявлено чотири ключові напрями застосування ШІ (генерація графічного контенту, автоматизація дизайну, текстова підтримка, UX-аналітика), які найбільше впливають на якість та швидкість веб-дизайну. Проведений аналіз може бути корисним інструментом для вибору оптимальних технологій, що сприятиме підвищенню якості дизайну, автоматизації процесів та покращенню користувацького досвіду.

Список літератури

1. 10Web AI Website Builder. Офіційний сайт URL: *https://10web.io (дата звернення 28.02.2025).*

2. Adobe Sensei. Офіційний сайт. URL: https://www.adobe.com (дата звернення 28.02.2025).

3. Глінська А. О. Використання штучного інтелекту у веб-дизайні / А. О. Глінська; наук. кер.: М. В. Колосніченко, Т. В. Струмінська // Вісник Київського національного університету технологій і дизайну. 2022. Вип. 1. С. 124-128. URL: *https://er.knutd.edu.ua/bitstream/123456789/22767/1/Innovatyka2022_V1_P124-128.pdf* (дата звернення 28.02.2025).

4. Hostinger AI Heatmap. Офіційний сайт. URL: https://www.hostinger.com (дата звернення: 28.02.2025).

5. Козачок А. О., Хмелівський Ю. С**.** Використання штучного інтелекту у веб-дизайні // Наукові записки Вінницького національного технічного університету. 2023. Вип. 3. С. 131–134.URL:*https://jvestnik-sss.donnu.edu.ua/article/view/15811/15713* (дата звернення 28.02.2025).

УДК 004.415.2:004.8

О. П. Доренський, В. А. Заріцкий
dorenskyiop@kntu.kr.ua, viktorzarickiy@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

УДОСКОНАЛЕНА КОНЦЕПТУАЛЬНА МОДЕЛЬ ТЕХНІЧНИХ ПРОЦЕСІВ ІНЖЕНЕРІЇ ПРОГРАМНИХ СИСТЕМ

У праці [1], ґрунтуючись на ДСТУ ISO/IEC/IEEE 15288:2016 та ДСТУ ISO/IEC/IEEE 12207:2018, на концептуальному рівні синтезовано модель технічних процесів інженерії програмних систем. Проте вона не враховує можливості технологій штучного інтелекту, які активно інтегруються в життєвий цикл програмного забезпечення ІТ-проєктів. Тож, це дослідження присвячене удосконаленню моделі [1] шляхом впровадження засобів / інструментів ШІ в процеси аналізу бізнесу або місії, визначення потреб і вимог стейкголдерів, визначення вимог до системи, визначення архітектури, проєктування робочого проєкту, аналізу системи, реалізації, інтеграції, верифікації, впровадження, валідації.

Аналіз потенціалу ШІ для етапів життєвого циклу програмних систем, активностей команди ІТ-проєктів [2], показав, що, враховуючи дослідження [3], відчутний ефект для інженерії програмних систем може забезпечити впровадження технологій ШІ у процеси: визначення вимог до системи, який закономірно впливає на успіх всього проєкту; аналіз системи – моделювання та прогнозування поведінки системи дозволяє виявляти потенційні проблеми на ранніх стадіях ЖЦ ПЗ; реалізація – генерація коду та інтелектуальні інструменти розробки підвищують продуктивність розробників та покращують якість коду, знижуючи кількість помилок і дефектів ПЗ; верифікація – автоматизоване тестування з використанням ШІ дозволяє підвищити покриття коду тестами при менших витратах ресурсу, що істотно впливає на якість продукту; супровід – аналітика та автоматична діагностика проблем підвищують якість підтримки та знижують експлуатаційні витрати. Удосконалену модель [1] представлено на рис. 1.

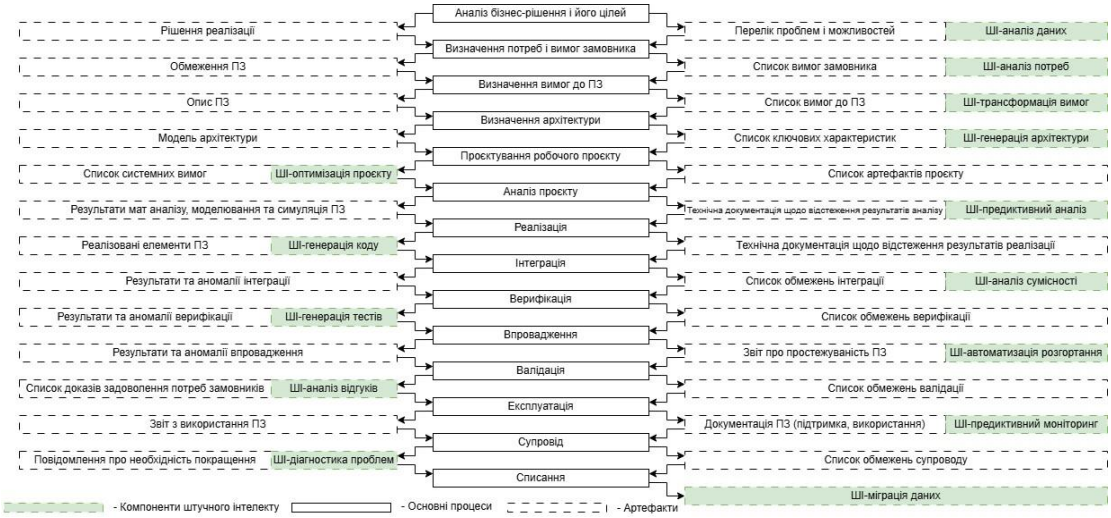


Рис. 1. Удосконалена концептуальна модель технічних процесів інженерії програмних систем

Загалом, інтеграції ШІ в технічні процеси інженерії ПС має потенціал для трансформації левової частини процесів ЖЦ ПЗ. ШІ здатний автоматизувати рутинні завдання, покращити якість рішень на кожному етапі реалізації ІТ-проєкту. Перспективою подальших розвідок є розроблення методик інтеграції інструментів ШІ, оцінювання ефективності впровадження запропонованих рішень.

Список літератури

1. Доренський О. П., Константинов О. Б. Концептуальна модель технічних процесів інженерії програмних систем. *Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі* : IX Міжна. наук.-практ. конф., 25-26 квіт. 2024 р., м. Київ : [матеріали конф.]. К.: КНУКіМ, 2024. С. 34-35.

2. Доренський О.П. Інформаційна модель вибору методології управління життєвим циклом програмного забезпечення інфотелекомунікаційних систем. Сучасні інформаційно-телекомунікаційні технології : матеріали наук.-техн. конф., 17–20 лист. 2015 р., Київ. К.: ДУТ, 2015. С. 114-116.

3. Kachurivskyi V., Kotovskyi A., Lykhodid T., Kachurivska H., Dorenskyi O. The Concept of Digital Transformation of Monitoring Scientific Activity of Participants in Educational Process of the Ukrainian HEI. *Central Ukrainian Scientific Bulletin. Technical Sciences*. 2025. Issue 11(42), Part I. P. 27-36.

УДК 004.4

Д.О. Гребенюк, *ст. 2-го курсу*
denis.hrebeniuk2006@gmail.com
Науковий керівник: Л.В. Константинова, *викладач*
Центральноукраїнський національний технічний університет, Кропивницький

ОГЛЯД ТА ПОРІВНЯННЯ ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ІНТЕРФЕЙСІВ МОБІЛЬНИХ ДОДАТКІВ НА ANDROID

На даний час мобільні застосунки стали важливою частиною життя людини. За статистикою понад 71% мобільних пристроїв на ринку використовують Android в якості основної операційної системи (ОС) [1]. Крім того, у 2025 році близько 3.3 мільярдів людей використовують цю ОС на своїх гаджетах [1]. Проте, зовнішній вигляд, зручність та швидкодія інтерфейсів дуже важливі при розробці сучасних мобільних додатків, і про це свідчить статистика: 28% користувачів видаляють встановлений додаток протягом 30 днів, якщо їх не задовільнила зручність дизайну або функціональність застосунку [2]. І тому для розробників огляд та порівняння інструментів для створення інтерфейсів мобільних додатків на Android є актуальним завданням на даний час.

У контексті розробки додатків для операційної системи Android існує два найпопулярніших інструменти для створення інтерфейсів: XML та Jetpack Compose. Обидва інструменти використовуються у різних рішеннях, але у них є свої переваги та недоліки.

Формат XML був вперше представлений у 1998 році, а з 2008 року його почали активно використовувати для проєктування інтерфейсів в Android. Він став невід'ємною частиною традиційної архітектури Android-додатків, що дозволяє відокремити візуальний шар від бізнес-логіки. Поява інтегрованого середовища розробки Android Studio значно спростила використання XML у розробці завдяки появі зручного графічного інструменту для роботи з ним, включаючи можливість перегляду інтерфейсу в реальному часі. Google постійно покращувала роботу з XML, зокрема у 2016 році представила потужний менеджер макетів ConstraintLayout, який дозволяє створювати складні й адаптивні інтерфейси з меншою кількістю вкладених елементів. Це не лише оптимізує продуктивність, а й робить код більш читабельним і гнучким, особливо у випадках підтримки багатьох розмірів екранів і орієнтацій [3].

Jetpack Compose був вперше анонсований Google у 2019 році як основне рішення для розробки складних та сучасних інтерфейсів для Android-додатків. Його основною метою є впровадження декларативного стилю програмування, при якому розробник описує, яким має бути інтерфейс, а не яким чином його будувати покроково. Подібний підхід вже успішно використовується у таких фреймворках, як Flutter, SwiftUI, Xamarin та React Native. Після випуску стабільної версії у липні 2021 року, Jetpack Compose почав активно впроваджуватись як у стартапах, так і у великих компаніях. Серед компаній, що впровадили Compose, - X (Twitter), Reddit, Dropbox та інші. Однією з ключових переваг є можливість значно пришвидшити розробку та тестування інтерфейсів завдяки зменшенню кількості шаблонного коду та покращеній інтеграції з Kotlin. У 2021 році було презентовано фреймворк Compose Multiplatform, який дозволяє створювати кросплатформені застосунки не лише для Android, а й для десктопів, вебу, а з 2023 року - і для iOS. Це відкриває нові можливості для універсальних інтерфейсів і значно спрощує підтримку проєктів, орієнтованих на декілька платформ одночасно [4][5].

Jetpack Compose та XML-візуалізації представляють два підходи для створення інтерфейсів мобільних додатків на платформі Android. Jetpack Compose базується на декларативному підході, де розробник описує бажаний стан інтерфейсу, а система автоматично забезпечує його рендеринг; натомість XML-візуалізації використовують імперативний підхід, який потребує ручного керування ієрархією компонентів та станами. Jetpack Compose дозволяє скоротити кількість шаблонного коду завдяки використанню Kotlin-функцій для створення інтерфейсних компонентів, тоді як XML-подання зазвичай включають окреме визначення інтерфейсу в XML-файлах із додатковою логікою на Java або Kotlin. Щодо попереднього перегляду змін, Jetpack Compose пропонує перегляд інтерфейсу та його інтерактивність у реальному часі, тоді як XML-візуалізації вимагають запуску додатка для повноцінного тестування інтерактивності. Jetpack Compose застосовує принципи реактивного програмування, що спрощує керування станами та оновлення інтерфейсів, на відміну від XML, де оновлення та зміни стану зазвичай реалізуються вручну. Крім того, Jetpack Compose містить вбудовану підтримку простіших способів створення анімацій та переходів, тоді як XML вимагає додаткових ресурсів і файлів налаштувань. З погляду продуктивності, Compose використовує оптимізований рушій рендерингу, що може зменшити кількість зайвих оновлень порівняно з XML-поданнями. В контексті підтримки та подальшого розвитку, Jetpack Compose входить до офіційної екосистеми Android Jetpack і активно розвивається Google, тоді як XML-візуалізації залишаються підтримуваними, але поступово отримують менше інноваційних оновлень [6].

Враховуючи все це, для створення інтерфейсів мобільних додатків на Android доцільно розглянути короткий опис та порівняння наданих інструментів (табл.1).

Таблиця 1

Коротке порівняння XML та Jetpack Compose [3].

Критерії порівняння	XML	Jetpack Compose
Синтаксис	Багатослівний, більш імперативний та великий об'єм коду	Декларативний, програмний та лаконічний
Мова	Java або Kotlin	Лише Kotlin
Структура коду	Кожен макет має окремий XML-файл, іноді складний в обслуговуванні	Вбудований код інтерфейсу в Kotlin, простий в обслуговуванні
Реактивність	Відсутня	Присутня
Складність	Простий у вивченні	Легше вивчити, ніж XML
Налаштування	Налаштовуваний	Більш гнучкий у налаштуванні, ніж XML
Використання	Широко використовується в старих проєктах	Широко використовується в нових проєктах
Спільнота	Широка	Стрімко зростає
Розробник	World Wide Web Consortium (W3C)	Google

У підсумку можна відмітити, що XML та Jetpack Compose є двома популярними інструментами для створення інтерфейсів мобільних додатків на платформі Android, кожен із яких має свої особливості, сферу використання, переваги та недоліки. XML є перевіреним часом і широковживаним рішенням з усталеною підтримкою та зручними графічними інструментами, натомість Jetpack Compose представляє сучасний декларативний підхід із перспективами кросплатформного розвитку та активною підтримкою з боку Google. Вибір конкретного інструменту залежить від потреб проєкту, особливостей розробки, команди розробників та стратегічних планів компанії.

Список літератури

1. Kumar N. Android Usage Statistics 2025: Devices & Market Share. DemandSage. URL: https://www.demandsage.com/android-statistics/ (дата звернення: 14.04.2025).
2. Kumar A. The Impact of UI/UX Design on Mobile App Retention Rates | TechTose. TechTose. URL: https://techtose.com/latest-insights/the-impact-of-ui-ux-design-on-mobile-app-retention-rates-techtose (дата звернення: 14.04.2025).
3. Angelo A. XML vs. Jetpack Compose: A Comparison. OpenReplay. URL: https://blog.openreplay.com/xml-vs-jetpack-compose--a-comparison/ (дата звернення: 30.03.2025).
4. Davenport C. JetBrains releases Compose Multiplatform 1.0 for creating Kotlin-based Android, desktop, and web apps. XDA. URL: https://www.xda-developers.com/jetbrains-releases-compose-multiplatform-1-0-for-creating-kotlin-based-android-desktop-and-web-apps/ (дата звернення: 30.03.2025).
5. Krill P. JetBrains adds iOS support to cross-platform UI framework. InfoWorld. URL: https://www.infoworld.com/article/2338526/jetbrains-adds-ios-support-to-cross-platform-ui-framework.html (дата звернення: 30.03.2025).
6. Vora I. Jetpack Compose vs XML: A comprehensive comparison for Android UI development. Aubergine Solutions. URL: https://www.aubergine.co/insights/jetpack-compose-vs-xml-a-comprehensive-comparison-for-android-ui-development (дата звернення: 30.03.2025).

УДК 004.8:004.4

К.О. Горбенко, *ст. 2-го курсу*
jeydj555@gmail.com
Науковий керівник: Л.В. Константинова, *викладач*
Центральноукраїнський національний технічний університет, Кропивницький

ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ГЕНЕРАТИВНОГО ШІ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО КОДУ

При розробці програмного забезпечення часто виникає низка проблем, таких як проблеми якості, вартості, часозатратності та ін. Пошук способів уникнути цих проблем - актуальна задача на даний час. Тому дослідження генеративного штучного інтелекту (ШІ) у програмуванні буде доцільним і спрямоване на оцінку його здатності підвищити продуктивність і скоротити час розробки. Актуальність зумовлена потребою в автоматизації рутинних завдань і підтримці розробників у створенні коду. Мета - з'ясувати, наскільки ефективно генеративний ШІ може вирішувати типові завдання програмування.

Генеративний ШІ аналізує великі масиви коду і здатен створювати нові фрагменти на основі текстового опису, що спрощує процес розробки. Принцип роботи генеративного ШІ базується на навчанні моделей на великих обсягах коду. Під час навчання ШІ вивчає синтаксис, структуру та логіку мов програмування, що дозволяє йому створювати функціональні фрагменти коду. Сучасні інструменти, як-от GitHub Copilot, ChatGPT, CodeWhisperer, Codeium активно застосовуються розробниками.

GitHub Copilot інтегрується з популярними середовищами розробки, такими як Visual Studio Code та IntelliJ, і пропонує автозаповнення коду на основі контексту. Дослідження GitHub показують, що використання Copilot може зменшити час виконання завдань на 55% [1]. ChatGPT використовується для генерації фрагментів коду, пояснення алгоритмів, написання SQL-запитів, налагодження помилок і створення документації. Він також підтримує багатокроковий діалог і здатний аналізувати великі обсяги коду. Amazon CodeWhisperer спеціалізується на інтеграції з AWS та генерації інфраструктурного коду. Він надає рекомендації на основі коментарів у природній мові та існуючого коду, що дозволяє розробникам швидше завершувати завдання [2]. Codeium забезпечує інтелектуальне автозаповнення, генерацію коду та пояснення, підтримуючи понад 70 мов програмування. Він інтегрується з більш ніж 40 редакторами, включаючи Visual Studio Code, Vim та Jupyter [3]. Усі ці інструменти активно використовуються в frontend-, backend-, мобільній, системній розробці, автоматизованому тестуванні та DevOps.

Дослідження впливу генеративного (ШІ) на продуктивність розробників демонструють значне підвищення ефективності в різних аспектах розробки програмного забезпечення. Зокрема, дослідження компанії SoftServe, проведене у 2023 році [4], охопило понад 1000 співробітників із семи країн. Учасники були поділені на експериментальну групу, яка використовувала моделі GPT-3.5/4.0, та контрольну групу без доступу до ШІ. Результати 1500 експериментів показали, що використання генеративного ШІ призвело до зменшення часу виконання завдань на 31% та підвищення загальної продуктивності команд до 45%.

Детальніше, вплив ШІ на продуктивність варіювався залежно від ролі в команді. Наприклад, у розробці вимог ШІ дозволяє швидше формулювати технічні завдання, у результаті чого продуктивність зростає на 44%. Для архітектурного дизайну ШІ допомагає оптимізувати структуру програмних систем, що збільшує ефективність на 39%. У роботі розробників він може згенерувати фрагменти коду та автоматично виправляти помилки, що призводить до підвищення продуктивності на 48%. Водночас інженери з контролю якості можуть значно скоротити час на тестування за допомогою автоматизованих генерацій тестів, що збільшує ефективність на 62% (табл.1). Усі ці фактори разом призводять до загального зменшення часу виконання завдань на 31% та підвищення загальної продуктивності команд на 45% [4].

Таблиця 1

Вплив генеративного ШІ на продуктивність за ролями в команді [4].

Роль у команді	Застосування ШІ	Підвищення продуктивності (%)
Бізнес-аналітики	Формулювання технічних завдань	44%
Архітектори ПЗ	Оптимізація структури систем	39%
Розробники	Генерація коду, виявлення помилок	48%
Фахівці з тестування	Автоматичне створення тестів	62%

Попри численні переваги, існують і виклики. Згенерований код потребує перевірки, може містити вразливості або не відповідати стандартам. Дослідження компанії Uplevel показало, що використання GitHub Copilot може призвести до збільшення кількості помилок у коді. Зокрема, розробники, які використовували Copilot, мали на 41% більше помилок у порівнянні з тими, хто не використовував цей інструмент [5]. Це свідчить про те, що інструменти ШІ не завжди враховують специфічні вимоги або контекст проєкту, що може призвести до помилок, які потребують додаткової перевірки та виправлення.

Згідно з опитуванням Stack Overflow 2023 року, 82% розробників-початківців (тих, хто навчається програмуванню) вже використовують або планують використовувати інструменти ШІ у своєму процесі розробки, порівняно з 70% професійних розробників. Це свідчить про високу зацікавленість молодих спеціалістів у впровадженні новітніх технологій[6] (табл.2). Водночас дослідники застерігають, що новачки можуть частіше покладатися на згенеровані рішення без належного розуміння їхньої логіки, що потенційно може призвести до некоректного використання або зниження якості коду.

Разом з тим, дослідження демонструють, що більш досвідчені розробники отримують вищу користь від використання ШІ. Так, згідно з аналізом компаній Zinnov та Ness Technologies, впровадження генеративного ШІ сприяє скороченню часу виконання завдань в середньому на 40%. При цьому цей показник був вищим серед фахівців середнього рівня (41%) та старших розробників (48%), ніж серед молодших спеціалістів (35%) [7]. Це може бути пов'язано з глибшим розумінням процесів розробки та вмінням ефективніше інтегрувати інструменти ШІ у робочий процес.

Таблиця 2

Порівняльна характеристика інструментів генеративного ШІ для програмування.

Інструмент	Основна функція	Інтеграція	Особливості
GitHub Copilot	Автозаповнення коду	Visual Studio Code, IntelliJ	Контекстуальні підказки, прискорення кодування на 55%
ChatGPT	Генерація коду, пояснення, налагодження	Вебінтерфейс, API	Підтримка багатокрокових діалогів, аналіз великих обсягів коду
CodeWhisperer	Генерація інфраструктурного коду	Інструменти AWS	Рекомендації на основі природної мови та коду
Codeium	Автозаповнення, генерація, пояснення коду	40+ редакторів (VS Code, Vim тощо)	Підтримка 70+ мов програмування

Отже, можна стверджувати, що генеративний штучний інтелект відкриває нові горизонти в автоматизації розробки програмного забезпечення, зменшуючи часові витрати та підвищуючи продуктивність команд. Його ефективність особливо відчутна серед досвідчених розробників, які здатні інтегрувати ШІ в робочий процес із врахуванням контексту і технічних вимог. Водночас масове впровадження таких технологій супроводжується викликами - від ризиків помилок і вразливостей до залежності менш досвідчених користувачів від згенерованих рішень. У цьому контексті важливо забезпечити поєднання технологічних інновацій із професійною підготовкою, критичним аналізом результатів генерації та дотриманням стандартів якості програмного забезпечення.

Список літератури

1. Measuring Impact of GitHub Copilot. GitHub Resources. URL: https: //resources.github.com/learn/pathways/copilot/essentials/measuring-the-impact-of-github-copilot/ (дата звернення: 15.04.2025).

2. How Accenture is using Amazon CodeWhisperer to improve developer productivity | Amazon Web Services. Amazon Web Services. URL: https://aws.amazon.com/ru/blogs/machine-learning/how-accenture-is-using-amazon-codewhisperer-to-improve-developer-productivity/ (дата звернення: 15.04.2025).

3. Raychura N. Codeium: A Free AI-Powered Toolkit for Developers. Medium. URL: https://medium.com/@raychuranirav/codeium-a-free-ai-powered-toolkit-for-developers-c960a6ad528d (дата звернення: 15.04.2025).

4. ШІ від OpenAI збільшує продуктивність розробників на 45%: дослідження – AIN. Новини IT, бізнесу та стартапів в Україні – AIN.ua. URL: https://ain.ua/2023/10/06/shi-vid-openai-zbilshuye-produktyvnist-rozrobnykiv-na-45-doslidzhennya/ (дата звернення: 15.04.2025).

5. Adarlo S. There's a Problem WIth AI Programming Assistants: They're Inserting Far More Errors Into Code. Futurism. URL: https://futurism.com/the-byte/ai-programming-assistants-code-error (дата звернення: 15.04.2025).

6. Stack Overflow Developer Survey 2023. Stack Overflow. URL: https://survey.stackoverflow.co/2023/ (дата звернення: 15.04.2025).

7. Gen AI widens productivity gap between senior and junior developers. Techcircle. URL: https://www.techcircle.in/2024/02/15/generative-ai-increases-productivity-gap-between-senior-and-junior-developers-report (дата звернення: 15.04.2025).

УДК 004.032.26

І.В. Варченко, Є.В. Мелешко
mrcrazyyt67@gmail.com, elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький, Україна

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ АНАЛІЗУ ТЕКСТІВ НА НАЯВНІСТЬ ІНФОРМАЦІЙНО-ПСИХОЛОГІЧНИХ ВПЛИВІВ

У сучасному інформаційному середовищі значну загрозу становить поширення дезінформації та контенту, що має прихований психологічний вплив на аудиторію. Такі матеріали часто використовують маніпулятивні мовні конструкції, впливають на емоційний стан читача та викривлюють сприйняття подій. У зв'язку з цим актуальним є створення інструментів, здатних автоматично виявляти ознаки інформаційно-психологічного впливу в текстах.

Метою цієї роботи є розробка програмного забезпечення системи штучного інтелекту, здатного аналізувати тексти та виявляти ознаки інформаційно-психологічного впливу.

Програмну реалізацію системи побудовано з використанням мови C# – для розробки алгоритмів, що обчислюють ключові лінгвістичні характеристики тексту, Python – для створення та навчання штучної нейронної мережі, а також середовища Unity – для реалізації графічного інтерфейсу користувача. Використовувалися бібліотеки TextBlob, VADER, Python.NET, TensorFlow, NumPy та Keras.

У процесі розробки було реалізовано систему, що використовує методи обробки природної мови (NLP) та алгоритми машинного навчання. Для оцінки текстів використовуються такі характеристики, як сентимент-анліз, аналіз маніпулятивної мови, оцінка лексичного розмаїття та аналіз суб'єктивності.

Сентимент-аналіз – визначає емоційне забарвлення тексту (позитивне, негативне або нейтральне). Реалізовано з використанням бібліотек VADER, яка спеціалізується на коротких повідомленнях і базується на словниковому підході з урахуванням контексту, та TextBlob, що побудована на основі NLTK і Pattern і забезпечує базову лінгвістичну обробку тексту.

Аналіз маніпулятивної мови – виявляє слова і конструкції, які можуть свідчити про спробу вплинути на думку читача. Алгоритм базується на словнику типових маніпулятивних виразів. Цей словник охоплює десять тематичних категорій: узагальнення, заклики до дії, емоційні тригери, вплив на основі страху, маніпулятивні контрасти, помилкові аналогії, підміна понять, заклики до патріотизму, формування сумніву й невпевненості, а також створення відчуття терміновості чи тривоги. На практиці алгоритм здійснює аналіз тексту шляхом проходження по лексемах і обчислює кількість збігів зі словником. Отримане значення нормалізується на загальну кількість слів у тексті, внаслідок чого формується індекс маніпулятивності.

Оцінка лексичного розмаїття – обчислюється індексом type-token ratio (TTR), що визначає співвідношення унікальних слів до загальної кількості слів. Високе значення TTR свідчить про розмаїття лексики, тоді як низьке – про її повторюваність. Такий індекс може слугувати маркером автоматизованих або шаблонних текстів, а також текстів із потенційним маніпулятивним змістом.

Аналіз суб'єктивності – визначає, наскільки текст виражає суб'єктивну думку, реалізовано за допомогою TextBlob, яка дозволяє визначити рівень суб'єктивності за допомогою лексичних шаблонів і граматичних конструкцій. Отримане значення варіюється в діапазоні від 0 (повна об'єктивність) до 1 (максимальна суб'єктивність). Така характеристика є важливою при виявленні текстів, що апелюють до емоцій замість фактів, що є однією з ознак інформаційно-психологічного впливу.

Результати цих алгоритмів передаються до нейронної мережі, яка на їх основі виконує класифікацію тексту та формує підсумковий висновок.

Для класифікації текстів на наявність інформаційно-психологічного впливу було створено нейронну мережу з наступною структурою:

– Вхідний шар (Input Layer): складається з 4 нейронів (x1, x2, x3, x4), які отримують на вхід ознаки тексту: x1 – сентимент-анліз, x2 – аналіз маніпулятивної мови, x3 – оцінка лексичного розмаїття та x4 – аналіз суб'єктивності.

– Прихований шар: 8 нейронів (h1–h8) з функцією активації ReLU.

– Вихідний шар (Output Layer): складається з 2 нейронів: y1 – відповідає за прогнозований Label (класифікує текст за такими варіантами: Informative, Neutral, Objective, Persuasive, Positive, Subjective, Manipulative), y2 – відповідає за пояснення (Conclusion), що дозволяє сформулювати коротке текстове обґрунтування результату.

Навчання моделі здійснювалося на основі попередньо підготовленого датасету із застосуванням оптимізатора Adam та функції втрат categorical crossentropy.

Структура розробленої нейронної мережі зображена на рис. 1.

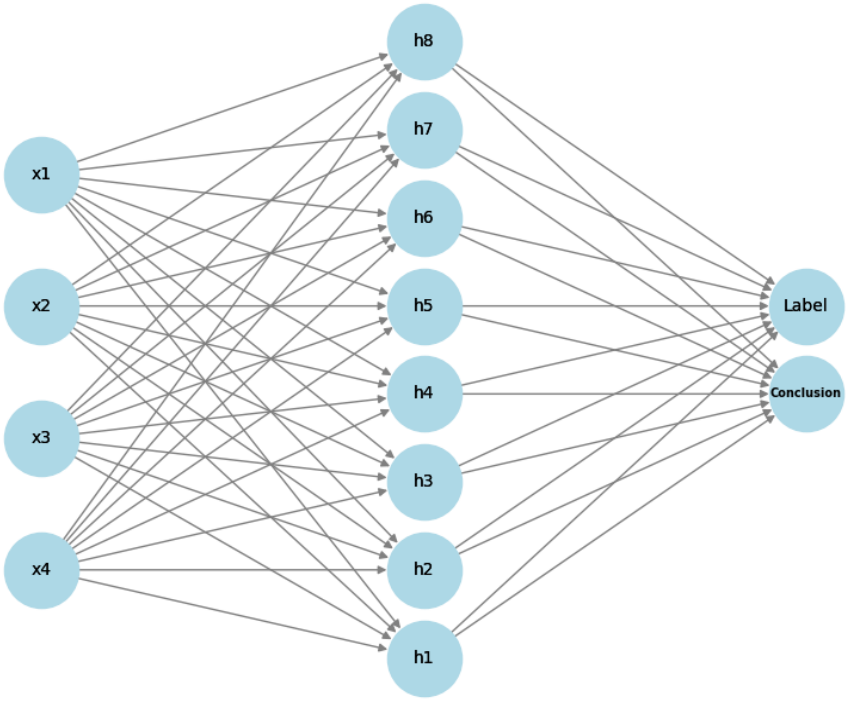


Рис. 1. Структура нейронної мережі

Таким чином, розроблене програмне забезпечення дозволяє користувачу вводити текст, аналізувати його на наявність інформаційно-психологічного впливу, отримувати результат у вигляді класифікації та короткого пояснення, а за потреби – зберігати висновки у текстовий файл. Система може використовуватись у сфері інформаційної безпеки, журналістиці, медіа-аналітиці та інших галузях, де потрібно швидко оцінювати великі обсяги текстів.

Програма побудована так, що за потреби її можна доповнювати – наприклад, розширити словник маніпулятивної лексики або адаптувати до інших мов. Це дозволяє використовувати її в ширшому контексті, зокрема, для моніторингу соціальних мереж чи інших інформаційних джерел.

Тож, у роботі було реалізовано метод автоматизованого виявлення інформаційно-психологічних впливів у текстах на основі нейронної мережі та попереднього аналізу сентименту, суб'єктивності, лексичного різноманіття і маніпулятивної мови.

Список літератури

1. Perkhach R.-Y. linguistic methods of manipulation in press: content analysis [Електронний ресурс] / R.-Y. Perkhach, A. Smyrnova // Young Scientist. – 2019. – Т. 10, № 74. – Режим доступу: https://doi.org/10.32839/2304-5809/2019-10-74-43
2. Type Token Ratio in NLP [Електронний ресурс] // Medium. – Режим доступу: https://medium.com/@rajeswaridepala/empirical-laws-ttr-cc9f826d304d
3. Detecting subjectivity and tone with automated text analysis tools [Електронний ресурс] // Pew Research Center. – Режим доступу: https://www.pewresearch.org/decoded/2018/07/26/detecting-subjectivity-and-tone-with-automated-text-analysis-tools/
4. Аналіз тональності тексту [Електронний ресурс] // Вікіпедія. – Режим доступу: https://uk.wikipedia.org/w/index.php?title=Аналіз_тональності_тексту&oldid=44485908
5. Guide | TensorFlow Core [Електронний ресурс]. – Режим доступу: https://www.tensorflow.org/guide

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 , matsuyan@ukr.net , elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

РОЗШИРЕНЕ ТЕСТУВАННЯ НА РОЗПОДІЛ ЧИСЕЛ ГЕНЕРАТОРА ПСЕВДОВИПАДКОВИХ ЧИСЕЛ НА ОСНОВІ БІЛЬЯРДА СІНАЯ

В роботі проводиться тестування генератора псевдовипадкових чисел (ГПВЧ) на основі більярда Сіная, який використовує математичні принципи, що базуються на траєкторії руху кулі у більярдному столі при генерації випадкових чисел. Цей метод полягає в тому, щоб визначити точки перетину кулі з границями столу (поля), які використовуються для генерації випадкових чисел [1]. Поле має форму опуклих кривих, а точка частинки рухається по більярду з постійною швидкістю. Коли вона досягає кордону поля більярда, зазнає пружного зіткнення з дзеркальним відображенням відповідно до закону відбиття – кут падіння дорівнює куту відбиття. Між двома зіткненнями точка рухається прямолінійно. На рис 1 зображено візуалізацію руху математичної точки при 20-ти ітераціях.

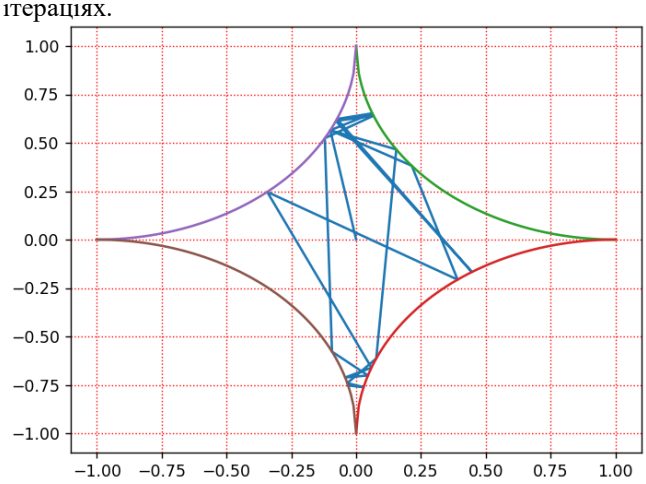


Рис.1 – Приклад візуалізації руху математичної точки в більярді Сіная при 20-ти ітераціях під час генерації ключа шифрування

З рис. 1 видно, що математичні частинки починають свій рух від центру поля в бік початкового кута, який задається на початку розрахунків. Псевдовипадкові числа вираховуються з координат відбиття точок від границь поля. В попередніх тестах координати перетворюються в двійкове число [2]. В попередніх тестах було достатньо двійкового числа, але для подальших тестів згенеровані числа були переформатовані у вигляд 0 – 255 [3]. Наразі число розраховується з координат, сумуючи їх та з множенням на число Пі і подальшим діленням по модулю на 255. За простим тестом з виведенням графіку (графік заповнюється максимально рівномірно) це дало найкращий розподіл чисел. Розрахунок кількості оригінальних значень - 255. Такий вигляд числа дозволяє повною мірою протестувати ГПВЧ та може використовуватись для генерації ключів шифрування [4]. Програма ГПВЧ прораховує 10 мільйонів чисел та записує їх у файл. Кожен тест це окрема програма, яка читає файл та видає результат тесту. Проводились наступні тести: Chi-square тест; тест на автокореляцію; тест на періодичність; тест Монте-Карло; тест Коля; тест Діковича-Флореса.

Chi-square тест - це статистичний тест, який використовується для перевірки гіпотез про розподіл даних. У контексті тестування (ГПВЧ), цей тест застосовується для перевірки рівномірності розподілу згенерованих чисел, тобто чи числа рівноймовірно потрапляють в кожен інтервал можливих значень. Результати тесту:

- Хі-квадрат статистика: 291.9947264;
- P-значення: 0.05544408848882821;
- немає підстав відхилити гіпотезу про рівномірний розподіл.

Тест на автокореляцію використовується для перевірки наявності кореляції між елементами послідовності випадкових чисел, згенерованих генератором псевдовипадкових чисел (ГПВЧ). Основна ідея полягає в тому, що випадкова послідовність не повинна мати помітних закономірностей між різними елементами на певних інтервалах. Тест допомагає виявити чи є залежність між значеннями у послідовності через певну кількість кроків. На рис.2 зображено графік результату тесту.

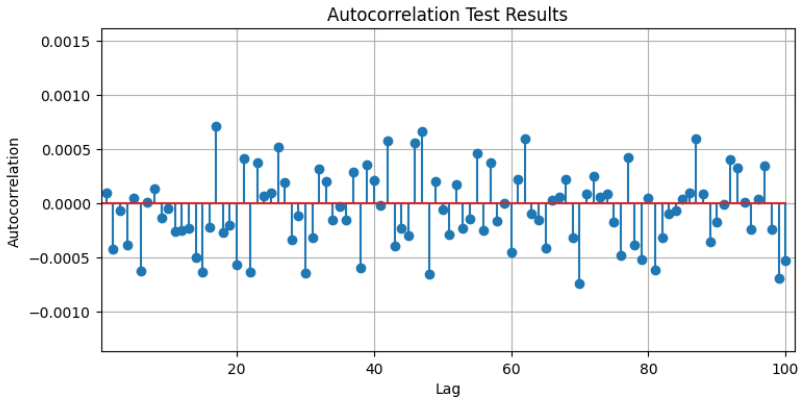


Рис.2 – Графік тесту на автокореляцію

Послідовність є випадковою оскільки не значно відрізняється від нуля через певний інтервал (лаг).

Тест на періодичність є важливим інструментом для оцінки якості генератора випадкових чисел. Якщо генератор має період, то він рано чи пізно почне повторювати одні й ті самі послідовності, що робить його непридатним для багатьох задач, де потрібна справжня випадковість. Тому цей тест допомагає виявити наявність регулярних повторень у згенерованих числах. Тест показав, що періодичність не знайдено.

Тест Монте-Карло використовується для перевірки якості (ГПВЧ), заснований на методах випадкових вибірок. Цей тест є прикладом використання методу Монте-Карло для статистичного тестування, де випадкові числа використовуються для моделювання певної задачі, результати якої можна порівняти з теоретичними значеннями. Якщо ГПВЧ добре працює, то результати моделювання будуть збігатися з теоретичними очікуваннями. У контексті тесту для ГПВЧ метод Монте-Карло використовується для перевірки рівномірності та якості випадкових чисел. Результат: Chi-square статистика: 291.9947264; Розподіл рівномірний на рівні значущості 5%.

Тест Коля використовується для виявлення послідовної кореляції (серійної кореляції) у ряді даних, зокрема у випадкових числах, що генеруються генератором псевдовипадкових чисел (ГПВЧ). Він допомагає перевірити, чи залежить кожне наступне випадкове число від попереднього. Ідеально, у випадковому ряді чисел не повинно бути кореляції між сусідніми значеннями — кожне нове число повинно бути незалежним від попередніх. Тест Коля допомагає перевірити, чи залежить кожне наступне випадкове число від попереднього. Результат: Статистика Коля (D): 0.00390625; P-значення: 1.0; розподіл є рівномірним на рівні значущості 5%

Тест Діковича-Флореса (Dikovich-Flores Test) є одним із статистичних тестів для оцінки якості генераторів псевдовипадкових чисел (ГПВЧ). Його мета - перевірка рівномірного розподілу згенерованих випадкових чисел на інтервалі, а також виявлення будь-яких закономірностей чи кореляцій, що можуть свідчити про невипадковий характер послідовності. Результат: P-значення: 0.0; Розподіл не є нормальним на рівні значущості 5%. Такий результат зумовлений особливістю генерації послідовності моделювання більярда та може сприйматись як закономірність.

Тестування ГПВЧ на основі більярда Сіная показала хороші результати для використання генерації ключів шифрування. Особливістю алгоритму ГПВЧ є те, що можлива генерація однакової послідовності псевдовипакових чисел на двох окремих пристроях. Тому використання алгоритму може використовуватись в системах захищених радіоканалів зв'язку.

Список літератури

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

2. Собінов О.Г. Простий генератор псевдовипадкової послідовності // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 груд. 2014 р. – Кіровоград: КНТУ, 2014. – С. 184.

3. Майданик О.О., Мелешко Є.В., Мацуй А.М. Тестування генератора псевдовипадкових чисел на основі більярда Синая. Матеріали XV Міжнародної науково-практичної конференції «Комплексне забезпечення якості технологічних процесів та систем» (23-24 травня 2024р., Чернігів). Чернігів: Національний університет "Чернігівська політехніка". Том 2. С 310 – 311.

4. Майданик О.О., Мелешко Є.В., Мацуй А.М. Генератор псевдовипадкових чисел для системи забезпечення захищеного радіоканалу зв'язку безпілотного літального апарату. Матеріали VI Міжнар. наук.-практ. конф. «Інформаційна безпека та комп'ютерні технології» (20-21 травня, 2023р., Кропивницький). Кропивницький: ЦНТУ, 2023. С 39 – 40.

УДК 004.4, 004.75, 004.8

Г.М. Дрєєва, О.М. Дрєєв
ganna@gmail.com, drey_sanya@ukr.net
Центральноукраїнський національний технічний університет, Кропивницький

РОЗПОДІЛЕНИЙ МОНІТОРИНГ НАТОВПУ

Часто для масових заходів використовують великі території зі складними зв'язками між майданчиками, наприклад, парки або території базарів та заводів. До прикладу таких заходів можна віднести святкування днів міста, фанкоми та інші подібні заходи, які супроводжуються одночасними виставками і виступами на декількох майданчиках, та іншими заходами. На жаль, бувають випадки, при проведені подібних заходів, утворення тисняви в результаті яких люди можуть отримати травми що можуть навіть мати летальний характер [1, 2]. Тому організаторам подібних заходів потрібно мати інформацію про пріоритетні місця накопичення натовпу з прогнозуванням небезпечної для утворення тисняви щільності людей на обмеженому просторі. Відповідно до доступності такої інформації, адміністрація має можливість затримати окремі виступи, організувати відволікаючі заходи на шляхах накопичення натовпу, перекривати шляхи або надавати доступ до резервних шляхів руху людей. Тому авторами було поставлено за мету отримати інформацію про рух натовпу та на основі цього руху прогнозувати кількість людей на окремих майданчиках на достатній, для організації запобіжних дій, час.

Отримання даних про рух натовпу більш просто отримати з панорамних камер спостереження, якими в більшості випадків обладнано місця проведення масових заходів. За для об'єктивності результатів спостереження авторами розроблено нейронну мережу з архітектурою U-Net, яка здатна виділяти ділянки з натовпом. Проведено ряд навчань нейронної мережі з різними кількостями шарів згортки та кількості фільтрів для обрання мінімальної складності нейронної мережі. Отримано нейронну мережу, яку здатен обробити мікрокомп'ютер за типом Raspberry Pi з частотою від 20 кадрів на секунду. При цьому нейронна мережа має лише 29601 параметрів, що істотно відрізняється від розповсюджених нейронних мереж для розмітки зображення [3, 4, 5, 6]. Результат роботи нейронної мережі можна оцінити на рис. 1:

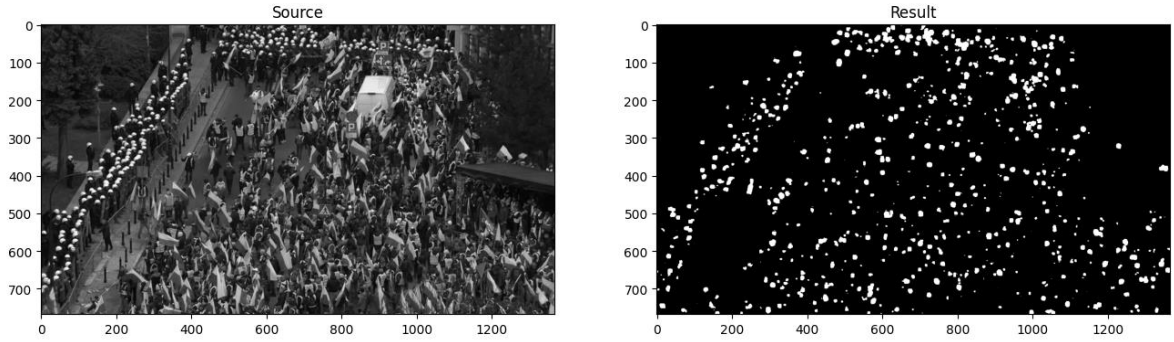


Рис. 1. Зображення кадру з натовпом та результатом виділення людських мас нейронною мережею

Завдяки мінімізації нейронної мережі, авторам вдалося покласти аналіз кадрів на самі комп'ютеризовані камери спостереження. Тому до серверу відправляються лише обмежені по обсягу інформації пакети, які містять оцінку щільності натовпу та пріоритетні напрямки руху натовпу. Це дозволяє використати GSM Інтернет зв'язок для моніторингу порівняно великої кількості камер без перевантаження інформаційної мережі. Також в якості серверу може виступати обчислювальна машина з невеликими обчислювальними потужностями, що значно здешевшує побудову та експлуатацію системи.

На отриману маску розташування людей, для послідовних кадрів шукається оптичний потік. Це дає розуміння про індивідуальні напрямки руху окремих груп людей. Відповідно до зони спостереження окремої камери визначається кількість людей, які прямують в напрямку суміжних зон спостереження, що і дає інформацію про динаміку розподілу людських мас по зонам спостереження.

Подібні розмітки проводяться на всіх камерах спостереження, які знаходяться на території. Відповідно до зон спостереження та шляхи між ними складають граф (рис. 2). Зони спостереження поділяються на внутрішні, які не мають зовнішніх від території проведення заходу притоків натовпу, та крайові – зона має вхід та вихід. Отримана топологічна модель на графі дозволяє побудувати ітераційну модель руху натовпу, де вхідний та вихідний потік приймається постійним по величині середнього значення на час прогнозування. В результаті кількома ітераціями будується послідовність чисельності людей на вузлах графу, якій зображає спостережну територію. Оператор може в реальному часі оцінювати прогнози, та маючі список доступних запобіжних заходів вчасно їх вжити.

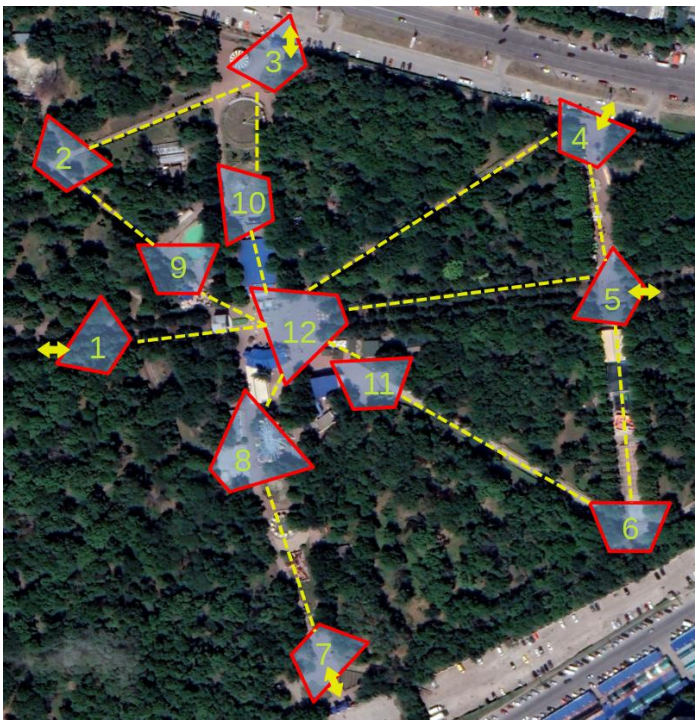


Рис. 2. Зони спостереження та шляхи людських потоків, які утворюють граф

Завдяки знайденим пріоритетним людським потокам та запізнення прибуття людей від зони до зони спостереження, система піддається прогнозуванню на 5-10 хвилин, що дає змогу задіяти заздалегідь підготовлені заходи відволікання та перенаправлення людських мас.

Висновки. В результаті дослідження авторами побудовано засобами ШІ мінімальну нейронну мережу для розмітки натовпу в кадрі отриманого з камери спостереження. Забезпечено розподільність обробки результатів малими комп'ютерними системами. Незалежність систем спостереження забезпечують легкість в масштабуванні, а також поступового розгортання системи та покращення покриття території спостереження. Розроблено систему створення графу руху людських мас, що дозволило будувати прогнози небезпечного підвищення щільності натовпу. Розроблена система призначена для вчасного впровадження заходів запобігання утворення тисняви, при умовах заздалегідь підготовлених адміністрацією проведення заходу прийомів відволікання та перенаправлення людських потоків. Розроблена система може бути вдосконалена на основі алгоритмів оптимізації на графах для вироблення рекомендацій виду методу корегування потоків людських мас.

Список літератури

1. Crowd Surges: Why They Happen URL: https://www.arnolditkin.com/blog/concert-injuries/crowd-surges-why-they-happen/

2. What Makes Crowds Dangerous? https://www.arnolditkin.com/blog/concert-injuries/what-makes-crowds-dangerous-/

3. P. Reisman; O. Mano; S. Avidan; A. Shashua, and other Crowd detection in video sequences URL: https://ieeexplore.ieee.org/abstract/document/1336357

4. N. N. A. Sjarif, S. M. Shamsuddin. S. Z. Hashim DETECTION OF ABNORMAL BE-HAVIORS IN CROWD SCENE: A REVIEW Int. J. Advance. Soft Comput. Appl., Vol. 4, No. 1, March 2012. ISSN 2074-8523; Copyright © SCRG Publication, 2012. URL: https://www.i-csrs.org/Volumes/ijasca/IJASCA07_Formatted_revised_4_1_2012.pdf

5. Software product: Crowd Detection Monitor People Capacity and Occupancy Levels. URL: https://senstar.com/products/video-analytics/crowd-detection/

6. Dushyant Kumar Singh, Sumit Paroothi, Mayank Kumar, Mohd. Aquib Ansari Hu-man Crowd Detection for City Wide Surveillance/ Procedia Computer Science. Volume 171, 2020, Pages 350-359 (https://doi.org/10.1016/j.procs.2020.04.036) https://www.sciencedirect.com/science/article/pii/S1877050920310012

УДК 004.75, 004.8

О.М. Дрєєв, Г.М. Дрєєва
drey_sanya@ukr.net, ganna@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

МЕТОД ВИДІЛЕННЯ ЗНАЧУЩИХ ОЗНАК ДЛЯ КЛАСИФІКАТОРА ЗОБРАЖЕННЯ

Сучасні нейронні мережі використовують технологію уваги [1]. Це засоби допомагають частину нейронної мережі перетворити на систему виділення ключової інформації з вхідного тексту, зображення або іншої інформації. Окреме виділення зон підвищеної уваги дає розуміння, що при навчанні нейронна мережа виділяє саме важливі для прийняття рішень деталі та елементи, хоч і контролювати це можна поки лише вручну візуально. Проте, за подібні переваги потрібно платити ускладненням структури нейронної мережі та значним збільшенням кількості вагових коефіцієнтів. Авторами пропонується використовувати для простих задач альтернативне визначення значимих елементів, які є ключовими для прийняття рішення нейронною мережею, зокрема нейронними мережами класифікації. Загальний вигляд нейронної мережі для класифікації має вигляд, який показано на рис. 1:

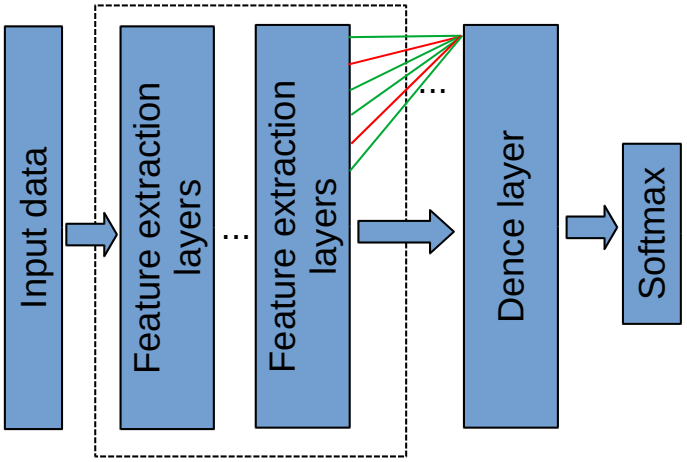


Рис. 1. Узагальнений вигляд нейронної мережі для класифікації

На рисунку вхідні дані підлягають аналізу різноманітними шарами, наприклад, для класифікації зображення це можуть бути згорткові шари та шари нормалізації. Це є варіативною частиною більшості нейронних мереж класифікації. Однак, як спільна риса таких мереж, є вихідний повнозв'язний шар, кількість виходів якого співпадає з кількістю визначених класів. Відповідно, кожен з вихідних нейронів має на вході додатні коефіцієнти та від'ємні коефіцієнти, що означає наявність відповідних ознак підсилює ймовірність належності зображення до відповідного класу, або послаблює цю ймовірність при від'ємному коефіцієнті. На основі цього, автори дослідження [1] для згорткової нейронної мережі запропонували метод визначення приблизної локалізації позитивних ознак належності зображення до класу. Проте, такий метод застосовний лише для нейронних мереж класифікації, які містять лише згорткові шари – ознаки з «віддалених» частин зображення не комбінуються. Тому, запропонований метод не працює на нейронних мережах типу Mixer [2].

Авторами запропоновано метод виділення важливих елементів зображення шляхом виродження вхідного зображення зі збереженням результату класифікації. Схематично такий процес показано на рис. 2.

На рис. 2 показано вхідне зображення, яке є ціллю оптимізаційного процесу, де результат класифікації досліджуваної нейронної мережі порівнюється з оригінальним виходом. Функцією втрат, в цьому випадку, є кількість наявних елементів на малюку, як результат середньоквадратичної суми результату фільтрації, яка залишає високочастотні зміни, та середньоквадратичне відхилення від початкового результату класифікації. Для стабілізації процесу можна вводити нормування проміжних результатів. В процесі оптимізації вхідного процесу конкурують дві величини. З однієї сторони, процес винищення переходів у зображенні, або вилучення деталей. З іншої сторони – збереження класифікації, або збереження елементів на зображенні, які відповідають за прийняття рішення про результат класифікації.

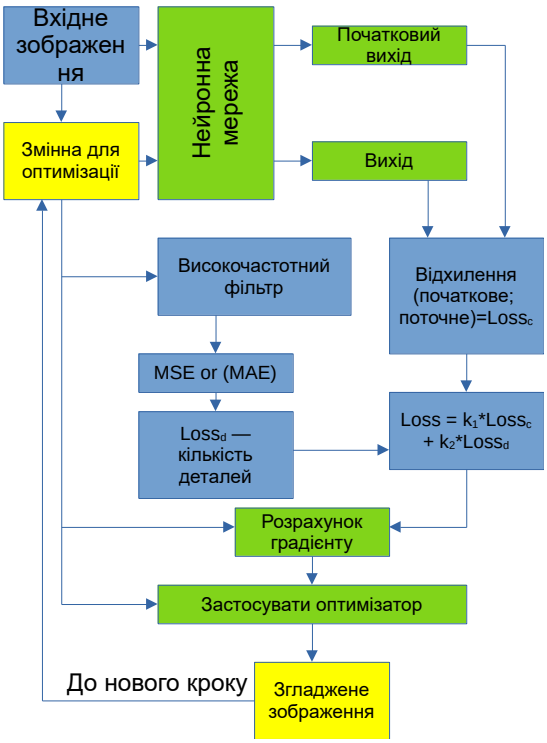


Рис. 2. Схема деградації зображення для виділення значущих ознак

Результат такого процесу деградації зображення зі збереженням результату класифікації можна побачити на рис. 3:

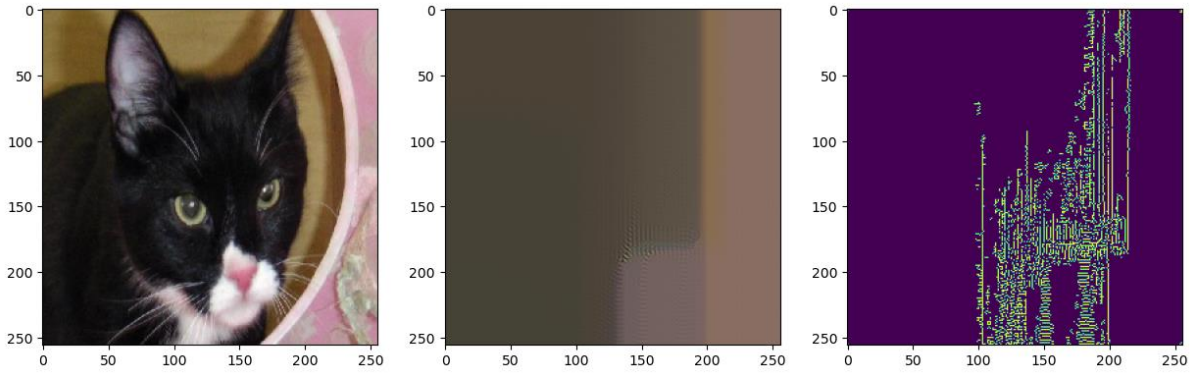


Рис. 3. Результат деградації зображення зі збереженням значущих елементів

Висновки. В результаті дослідження авторами побудовано систему контролю розподілення уваги навченої нейронної мережі для класифікації зображення. При використанні запропонованої методики, вхідне зображення (рис. 3 ліворуч) деградувало зі збереженням кольорового розподілення фону (рис. 3 центральне зображення). Високочастотні зони (рис. 3 праворуч), показують локалізацію дрібних елементів, які брали участь у формуванні результату класифікації. У разі не збігання збережених зон кольору та мілких елементів із об'єктом класифікації, говорить про те, що дані для навчання не є репрезентативними, і містять відмінності в залежності від класу на фонових об'єктах. Наприклад, фото котів значно частіше зустрічаються в домашніх умовах, а фото собак на фоні вуличних об'єктів.

Список літератури

1. Bolei Zhou, Aditya Khosla, Agata Lapedriza and other. Learning Deep Features for Discriminative Localization URL: https://cnnlocalization.csail.mit.edu/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf

2. Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov and other MLP-Mixer: An all-MLP Architecture for Vision URL: https://proceedings.neurips.cc/paper/2021/file/cba0a4ee5ccd02fda0fe3f9a3e7b89fe-Paper.pdf

УДК 004.52, 004.4

К.О. Задорожний, О.М. Дрєєв
zadorozhnyikostiantyn@gmail.com, drey_sanya@ukr.net
Центральноукраїнський національний технічний університет, Кропивницький

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИСТРОЮ ЗАПОБІГАННЯ ПРОСЛУХОВУВАННЮ

В задачі забезпечення безпеки інформації входять не лише захист цифрових даних. Також розглядаються фізичні канали витоку інформації, зокрема акустичні. Боротьба з прослуховуванням, як правило містить пошук та вилучення пристроїв прослуховування, але пасивні пристрої запису з швидкою передачею записаних розмов знайти вкрай важко. Наприклад, для пошуку таких закладних пристроїв використовують магнітометр, який вимірює магнітне поле від магнітного кріплення пристрою. Звісно, всі ці заходи не гарантують виявлення всіх закладних пристроїв. Тому також широко використовують методи, які ґрунтуються на створені завад передачі інформації.

Огляд існуючих пристроїв, показав, що наявні пристрої використовують мовоподібні шуми або ультразвукові шуми. Потужні ультразвукові коливання можуть сприйматися деякими автоматами регулювання вхідного підсилення, що призводить до малого підсилення звуку на вході пристрою звукозапису. Проте, надійність ультразвукових завад є занадто низькою, тому є сенс використовувати обидва методи одночасно. Відомі пристрої, які комбінують ці два методи боротьби з прослуховуванням, проте вони мало функціональні [1, 2]. Як спільні недоліки варто відмітити відсутність бази даних голосів. Тому, авторами запропоновано розробити власний пристрій, який би підтримував безконтактне керування та мав можливість використовувати базу даних записаних заздалегідь голосів. Корисно ввести автоматичне регулювання гучності мовоподібних завад.

В результаті проведеного аналізу, до пристрою виділено наступні вимоги: генерування гучного ультразвуку частотою 22КГц; програвання спотвореного звуку; запис зразку мовлення; здійснювати вибір між зразками записаних голосів; відображати поточний стан світловими індикаторами та надписами на екрані; керування пристроєм за http протоколом. Для забезпечення вказаних можливостей розроблено структурну схему пристрою (рис. 1) та алгоритм роботи розробленого програмного забезпечення, який показано на рис. 2:

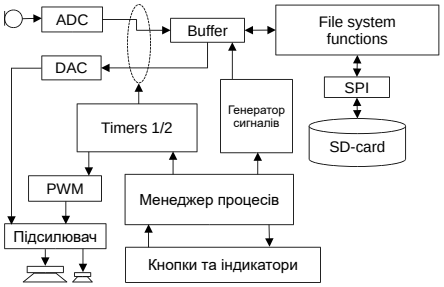


Рис. 1. Структурна схема пристрою перешкоджанню витоку інформації звуковим каналом

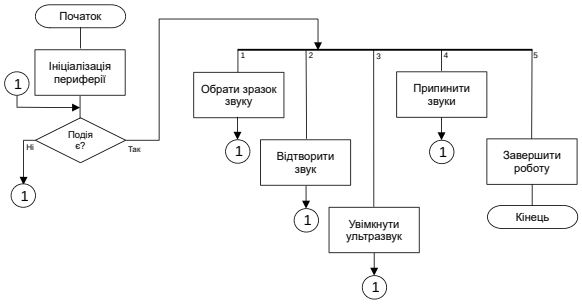


Рис. 2. Блок-схема алгоритму роботи програмного забезпечення пристрою

Висновки. В результаті огляду пристроїв створення завад для запобігання прихованого прослуховування, було виявлено не достатню функціональність наявних пристроїв, що до поставлених до них задач. Тому було сформовано вимоги до власного пристрою, які враховані при проектуванні та створенні програмного забезпечення. В результаті на базі мікроконтролера ESP32 побудовано пристрій для відтворення завад прослуховування розмов з ультразвуковими перешкодами та мовоподібними шумами на основі записаних зразків голосу.

Список літератури

1. В.В. Сінюгін, В.С. Катаєв, А.В. Грицак. Модульний генератор шуму для блокування витоку акустичної інформації. Вісник Вінницького політехнічного інституту. 2021. Вип. 6. С.158-164. DOI: https://doi.org/10.31649/1997-9266-2021-159-6-158-164

2. Моделювання шуму – білий, рожевий та коричневий шум, попси та тріски [Електронний ресурс]. URL: https://surl.li/hkyziz

УДК 004.52, 004.4

Д.С. Паращенко, Р.О. Лук'яненко, Г.М. Дрєєва
denus.parashencko@gmail.com, klank-07@ukr.net, ganna@gmail.com
Центральноукраїнський національний технічний університет, Кропивницький

СИСТЕМА ПРИХОВАНОЇ ПЕРЕДАЧІ ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ УЛЬТРАЗВУКОВОГО КАНАЛУ ВИТОКУ ІНФОРМАЦІЇ

Огляд сучасних пропозицій аналізу та протидії витокам інформації звуковими каналами показав, що наявні засоби досить часто уникають ультразвуковий діапазон. Тому авторами запропоновано пристрій передачі інформації ультразвуковим каналом з метою тестування можливостей витоку інформації. Для досягнення поставленої мети тестування ультразвукового каналу витоку інформації, було вирішено реалізувати пристрій передачі інформації. Наявність такого пристрою надає можливість продемонструвати важливість контролю ультразукового каналу витоку інформації, відпрацювати завадостійке кодування інформації та на практиці дослідити можливості надійності та швидкості передачі інформації подібними методами.

З метою спрощення апаратної частини пристрою для його більш простого відтворення навіть в домашніх умовах, авторами було покладено на мікроконтролер задачу генерування ультразвуків заданої частоти та їх бінарну модуляцію. Для виконання цієї задачі було використані можливості таймерів генерувати PWM (або ШІМ – широтно імпульсна модуляція) з наповненням 50%. Створені імпульси подаються на випромінювач ультразвуку за допомогою транзисторного ключа. Повідомлення для передачі та контроль прийнятого пакету інформації проводиться за допомогою бездротового інтерфейсу http протоколом, який здатен підтримувати мікроконтролер на базі ESP32. Також, вбудована можливість завадостійкого кодування інформації, алгоритм якого можна обирати. Передбачена можливість обмеженого керування пристроями без використання бездротового інтерфейсу, що покращує універсальність розробленого комплексу прихованої передачі інформації акустичними каналами.

Структурна схема розробленого програмного забезпечення показано на рис. 1 для передавача та на рис. 2:

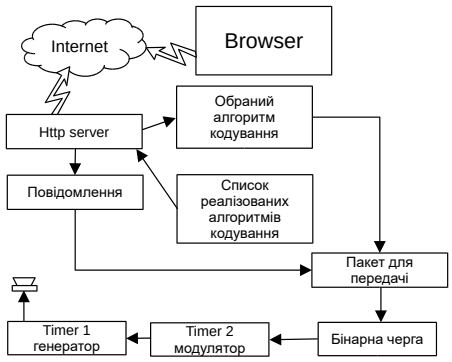


Рис. 1. Структурна схема програмного забезпечення передавача

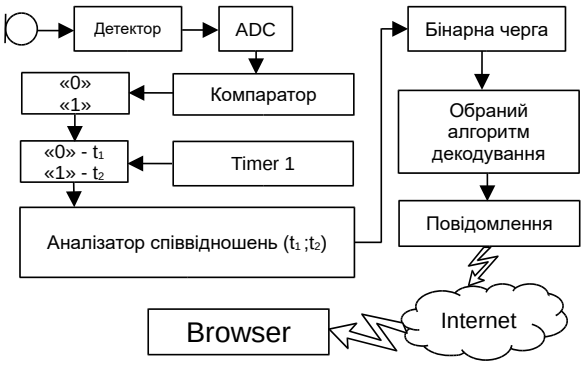


Рис. 2. Структурна схема програмного забезпечення приймача

Висновки. В результаті дослідження авторами побудовано апаратний комплекс передачі та прийому цифрової інформації з використанням ультразвуку. Для забезпечення роботи пристрою було розроблено програмне забезпечення за технологією інтернету речей для передавача та приймача інформації. Експериментально показано можливість прихованої передачі інформації, яку не можна зафіксувати без окремого аналізу ультразвукового фону. Доступність та відкритість програмного коду, дозволяє студентам відпрацьовувати алгоритми безпомилкової передачі інформації в умовах відсутності завад так і при їх високому рівні. Результати роботи пропонується використовувати в практичних заняттях вивчення акустичних каналів витоку інформації.

Список літератури

1. Ультразвук: застосування та процеси URL: https://www.hielscher.com/uk/technolo.htm/

2. J.R. Gonzalez, C.J. Bleakley "Low Cost, Wideband Ultrasonic Transmitter andReceiver for Array Signal Processing Applications" School of Computer Science and InformaticsUniversity College DublinBelfield, Dublin 4, Ireland. June 2011IEEE Sensors Journal 11(5):1284 – 1292 DOI: 10.1109/JSEN.2010.2084568

СЕКЦІЯ 3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ, МЕДИЦИНІ ТА ОСВІТІ

УДК 004.031 42

Т.Х. Фаталієв
tfataliyev@gmail.com
Інститут інформаційних технологій, Баку, Азербайджан

ДОСЛІДЖЕННЯ РОЛІ ШТУЧНОГО ІНТЕЛЕКТУ В РОЗВИТКУ ЦИФРОВОГО НАУКОВО-ОСВІТНЬОГО СЕРЕДОВИЩА

Сучасний період технологічного прогресу в суспільстві характеризується активною цифровізацією та впровадженням інтелектуальних систем, що поряд з іншими галузями, призводить до швидких трансформаційних процесів і в науці та освіті. Розвиток передових технологій Індустрії 4.0, таких як Інтернет речей (IoT), штучний інтелект (ІІ), кіберфізичні системи (КФС) та робототехніка відкриває нові можливості для автоматизації, підвищення ефективності та оптимального управління ресурсами. Ці технології сприяють якісній трансформації та прискоренню інтеграційних процесів у науці та освіті. В результаті формується нове корпоративне інноваційне середовище, що об'єднує е-науку та е-освіти [1,2]. В останні роки в цьому контексті методи ІІ та машинного навчання (МО) справили значний вплив на сфери науки та освіти. Ці методи створення моделей із наборів даних чи логічних алгоритмів, що імітують аспекти людської діяльності, стали ключовими інструментами для прискорення цифрового розвитку у цих галузях.

І так, з розвитком інформаційних та обчислювальних технологій автоматичні, адаптивні та ефективні технології ІІ стали широко застосовуватися в різних академічних галузях. ІІ в освіті (ІІВ) як міждисциплінарна область наголошує на застосуванні ІІ для сприяння навчальному процесу викладачів, розширення можливостей процесу навчання студентів та сприяння трансформації освітньої системи. По-перше, ІІО може покращити навчальний процес та педагогічний розвиток у процесах навчання, наприклад, автоматично оцінювати успішність учнів, контролювати та відстежувати навчання учнів та прогнозувати учнів, які перебувають у групі ризику. По-друге, ПІБ корисний для покращення навчання, орієнтованого на студентів шляхом надання адаптивного навчання, рекомендації персоналізованих навчальних ресурсів та діагностика прогалин у навчанні учнів. По-третє, ПІБ також відкриває можливості для перетворення освітньої системи, підкреслюючи важливу роль технологій, збагачуючи засоби передачі знань та змінюючи стосунки між викладачем та студентом.

Розвиток ПІБ також вніс зміни до області STEM (*science, technology, engineering and mathematics*) - освіти. STEM-освіта спрямована на поліпшення міждисциплінарних знань учнів та їх застосування, а також розвиток їх мислення вищого порядку, критичного мислення та здібностей вирішувати проблеми. Застосування ІІ у STEM-освіті має переваги, оскільки забезпечує адаптивні та персоналізовані навчальні середовища чи ресурси, а також допомогти викладачам зрозуміти поведінкові моделі навчання студентів та автоматично оцінити успішність у галузі. Широту охоплення та можливостей, створюваних додатками ІІ в цій галузі, можна представити в наступних категоріях, представлених в результаті лише одного дослідження [3]:

- *прогностичне навчання* – це попереднє прогнозування успішності або статусу навчання учнів з використанням алгоритмів ІІ та підходів до моделювання;
- *інтелектуальна система навчання* – яка визначається як система з підтримкою ІІ, яка була розроблена для надання індивідуальних інструкцій або зворотного зв'язку студентам та сприяння персоналізованому, адаптивному навчанню;
- *виявлення поведінки учнів* – системи, які відстежують та використовують навчальну поведінку, моделі та характеристики учнів за допомогою інтелектуального аналізу даних та аналітики навчання на основі ІІ у навчальних та освітніх процесах;
- *автоматизація* – технологія ІІ, яка дозволяє автоматично оцінювати успішність учнів та генерувати питання чи завдання для викладачів;
- *освітні роботи* – впровадження роботів у STEM-освіту для полегшення процесу навчання студентів, а також надання їм можливості набувати знань в інтерактивному режимі.

В даний час в системі науки та освіти існує безліч проблем, які потребують точності та терпіння для своєчасного вирішення завдань з мінімальними помилками. Тут ІІ бере на себе відповідальність за зниження критичної необхідності вирішення проблем та прийняття обґрунтованих рішень. Метою [4] є вивчення академічного та адміністративного застосування ІІ в освіті. Було показано, що додатки на основі ІІ не лише допомагають освіті в академічному та адміністративному плані, а й підвищують її ефективність. ІІ допомагає вчителям з різними типами завдань щодо формування та прийому учнів. Це зводить до мінімуму адміністративні обов'язки вчителя, дозволяючи йому більше вкладати у навчання та управління студентами. У [5] розглядається найважливіша роль ІІ в оптимізації адміністративних процесів, оптимізації розподілу ресурсів та персоналізації навчання. Інструменти на основі ІІ, такі як передиктивна аналітика, дозволяють виявляти студентів із групи ризику, прогнозувати тенденції зарахування та підтримувати стратегічне планування, тим самим покращуючи процеси прийняття рішень. Системи управління навчанням (LMS), оснащені можливостями

ІІ, пропонують персоналізований досвід навчання, сприяючи кращому залученню та індивідуальним освітнім траєкторіям для студентів. Крім того, автоматизовані адміністративні інструменти оптимізують такі завдання, як планування та управління ресурсами. що призводить до більш ефективної роботи та оптимального використання інституційних ресурсів.

Застосування ІІ у розвитку цифрового наукового середовища відрізняється різноманіттям [6]. На основі аналізу літератури в цій галузі можна навести як приклади такі узагальнені напрямки:

- *Обробка та аналіз великих обсягів даних* – обробка великих даних є однією з основних проблем у наукових дослідженнях. Алгоритми ІІ здатні ефективно аналізувати експериментальні дані, виявляти приховані закономірності та робити прогнози на основі статистичних моделей.
- *Моделювання та імітація* – використання ІІ для створення моделей складних систем, будь то фізичні та хімічні процеси, біологічні системи чи економічні моделі, відкрило нові горизонти у дослідженнях. Це дозволяє запускати моделювання, оптимізувати параметри та проводити експерименти у віртуальному середовищі, перш ніж впроваджувати їх у реальність. Саме тут важливу роль відіграють віртуальні лабораторії, автоматизовані лабораторії, які працюють у реальному часі.
- *Підтримка видавничої діяльності* – ІІ допомагає знаходити релевантні наукові публікації, автоматично складати огляди літератури та навіть писати чернетки статей, що прискорює процес підготовки наукових матеріалів.
- *Управління та менеджмент* – допомагає підвищити ефективність наукових досліджень, оптимізувати ресурси та покращити процеси прийняття рішень. На основі аналізу наукових публікацій, можна виявляти тренди та прогалини у дослідженнях. Моделі МО здатні передбачати, які наукові гіпотези чи напрями досліджень із великою ймовірністю дадуть успішні результати. ІІ може аналізувати потреби у фінансуванні, обладнанні та персоналі, допомагаючи ефективніше планувати проекти. ІІ може автоматично аналізувати цитованість, індекс Хірша та інші метрики, спрощуючи оцінку ефективності вчених та організацій.

Слід зазначити, що інтеграція штучного інтелекту в цифрове науково-освітнє середовище не тільки прискорює навчальні та дослідницькі процеси, але й формує основу для нових моделей взаємодії навчальних закладів, наукових організацій з науковцями, освітянами та студентами. ІІ грає також важливу роль в управлінні персоналом (УП), допомагаючи автоматизувати рутинні завдання, покращувати підбір, навчання та мотивацію співробітників. Незважаючи на існуючі виклики, такі як конфіденційність даних та етичні питання, ІІ продовжить впроваджуватися в УП, створюючи ефективніші та персоналізовані підходи до УП.

Резюмуючи, можна дійти невтішного висновку у тому, що використання додатків ІІ у сфері освіти та науки є важливим чинником цифрового розвитку цих структур. Вона спрямована на оптимізацію процесів навчання, дослідження, управління та обміну знаннями за допомогою цифрових рішень та представляє значний потенціал для підвищення ефективності, оперативності прийняття рішень, а також покращення процесів наукової та освітньої діяльності, включаючи управління.

Список літератури

1. Fataliyev T., Bayramov, H., Mikayilova R. Analysis and new approaches to solving the problems of integrating e-science and e-education based on the challenges of Industry 4.0. 5th International Conference on Problems of Cybernetics and Informatics (PCI 2023), Baku, Azerbaijan, August 28-30, 2023, p. 1-5.
2. Fataliyev T.Kh., Mehdiyev Sh.A., The impact of Industry 4.0 on the formation of Science 4.0, Problems of Information Technology, vol. 13, no. 2, pp. 37-45, 2022.
3. Ahmad, S.F., etc., Academic and Administrative Role of Artificial Intelligence in Education. Sustainability 2022, 14, 1101. https://doi.org/10.3390/su14031101
4. Olanike A.A., Enitan Sh.A., Njideka R.C., Integrating AI and technology in educational administration: Improving efficiency and educational quality. Open Access Research Journal of Science and Technology, 11(02), pp.116–127, 2024.
5. Xu W., Ouyang F. The application of AI technologies in STEM education: a systematic review from 2011 to 2021. IJ STEM Ed 9, 59 (2022). https://doi.org/10.1186/s40594-022-00377-5
6. Foresight: Use and impact of Artificial Intelligence in the scientific process. Luxembourg: Publications Office of the EU, 28 p., 2023. https://erc.europa.eu/sites/default/files/2023-12/AI_in_science.pdf

УДК 004.0, 004.6

K. Terletska[1], M. Prokopets[2]
khrystynaterletska1@gmail.com, maksymprokopets18@gmail.com
[1]*DraftKings Inc., Kyiv*
[2]*Lviv Polytechnic National University, Lviv*

AI-OPTIMIZED DATA LAKES IN HEALTHCARE DATA MANAGEMENT

Data lakes have become indispensable for managing large volumes of healthcare data, offering a centralized repository for various data types. These repositories can store structured data (e.g., patient records), semi-structured data (e.g., JSON files), and unstructured data (e.g., medical images, doctors' notes). The flexibility and scalability of data lakes [1] make them ideal for handling the extensive and diverse data generated in healthcare settings. By consolidating data from multiple sources into a single location, data lakes enable comprehensive data analysis and support advanced healthcare applications such as predictive analytics and personalized medicine.

Data lakes present significant challenges regarding efficient data retrieval and management despite their advantages. The complexity and scale of healthcare data, characterized by its volume, variety, and velocity, render traditional indexing and query optimization methods inadequate. Conventional indexing techniques, such as B-trees and hash tables, struggle to keep pace with the dynamic and heterogeneous nature of healthcare data. Similarly, traditional query optimization methods often fail to deliver the necessary performance improvements, leading to slow query responses and inefficient data access. These limitations hinder healthcare providers' ability to quickly and accurately retrieve critical patient information, impacting clinical decision-making and patient care.

This paper explores AI-driven solutions to address these challenges. The primary goal is to utilize AI to optimize data indexing and query processes, thereby improving the speed and accuracy of data retrieval in healthcare settings. By leveraging machine learning and deep learning algorithms, the proposed approach aims to enhance the performance of data lakes [2], making them more efficient and effective for healthcare data management. Specifically, the paper focuses on automating the indexing process to create more efficient data structures and optimizing query execution to reduce response times.

Enhanced data management capabilities can lead to better clinical decision-making, improved patient care, and streamlined healthcare operations. Efficient data retrieval allows healthcare providers to access critical patient information quickly, supporting timely and informed decisions. Additionally, optimized data lakes can reduce operational costs and improve the overall efficiency of healthcare organizations. By addressing the challenges of data management in healthcare [3], AI-optimized data lakes can transform how healthcare providers interact with and utilize data, ultimately leading to better health outcomes for patients.

The algorithm of AI-optimized data lakes for healthcare data management is illustrated in Figure 1. This AI-optimized approach significantly enhances data retrieval efficiency, enabling swift access to critical patient information and supporting timely clinical decisions. It improves clinical decision-making, reduces operational costs, and ensures high data quality.

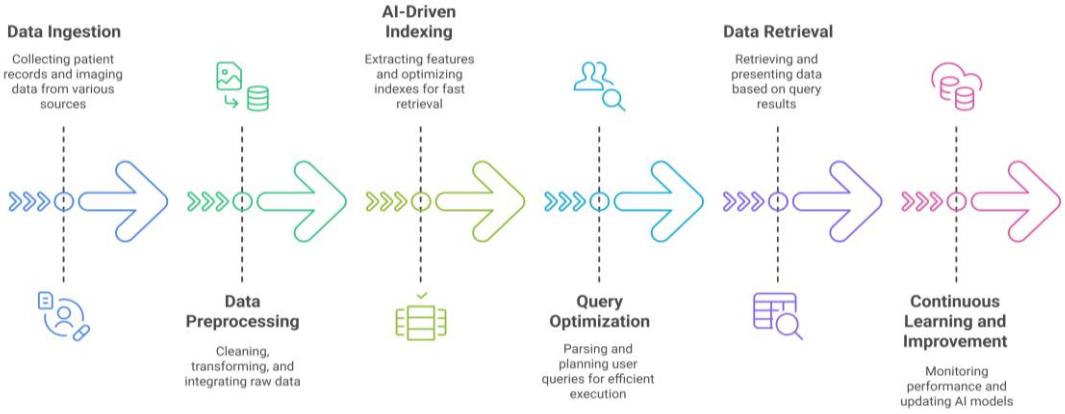


Fig. 1. The algorithm of AI-optimized data lakes in healthcare data management

The system's scalability and flexibility are increased, allowing it to manage growing data volumes and adapt to changing patterns. Continuous learning and improvement facilitate proactive healthcare management. This approach integrates diverse data sources, supports advanced analytics, and results in better patient care and outcomes.

Data ingestion involves collecting patient records, medical imaging data, and treatment histories from various sources such as Electronic Health Record systems and imaging devices. This raw data is then stored in the data lake, providing a centralized repository for diverse data types.

Data preprocessing is the next step, where raw data from the data lake undergoes cleaning to remove duplicates, handle missing values, and correct errors. The data is then transformed to normalize formats and convert data types, followed by integration to merge data from different sources into a unified dataset. This preprocessed data is now ready for indexing and analysis.

AI-driven indexing begins with feature extraction, identifying key features for indexing such as patient ID, date, and type of data. AI algorithms are used to create efficient indexes by recognizing patterns in the data. These indexes are continuously optimized based on query patterns and data access frequency, ensuring fast data retrieval.

Query optimization involves analyzing and parsing user queries to understand the data requirements. AI algorithms generate an optimized query execution plan, selecting the most efficient way to access and retrieve the required data. The optimized query is then executed on the data lake [4], leveraging the AI-driven indexes for fast and accurate data retrieval.

Data retrieval entails accessing the required data from the data lake based on the optimized query results. This involves presenting the retrieved data to the user in a usable format.

Continuous learning and improvement are essential for maintaining the efficiency of AI-optimized data lakes. Feedback from query performance and user interactions tracks query performance and data access patterns. AI models are retrained using this feedback to improve indexing and query optimization. Updated AI models are then deployed to the data lake system, ensuring continuous improvement in data retrieval efficiency.

In this case study, a large healthcare organization manages extensive patient records, medical imaging data, and treatment histories. The organization faces challenges in efficiently retrieving and handling this vast volume of data, impacting clinical decision-making and patient care. The primary goal is to leverage AI to enhance data indexing and querying processes, thereby improving the speed and accuracy of data retrieval. This will enable healthcare providers to access critical patient information, facilitating timely, quick, and informed clinical decisions.

The implementation involves several key steps. Data ingestion starts with collecting data from various sources such as EHR systems, imaging devices, and treatment histories, and storing it in the data lake. Data preprocessing follows, where the raw data is cleaned, transformed, and integrated to ensure it is ready for indexing and analysis. AI-driven indexing is then applied, using AI algorithms to create and optimize indexes based on key features such as patient ID, date, and data type. This includes continuous optimization based on query patterns and data access frequency.

Query optimization involves analyzing and parsing user queries, generating optimized query execution plans using AI algorithms, and executing the optimized queries on the data lake. Data retrieval entails accessing the required data from the data lake based on the optimized query results and presenting it to the user. Continuous learning and improvement are essential for maintaining the efficiency of AI-optimized data lakes. Feedback from query performance and user interactions is used to monitor query performance and data access patterns. AI models are retrained using this feedback to improve indexing and query optimization. Updated AI models are then deployed to the data lake system, ensuring continuous improvement in data retrieval efficiency.

The case study demonstrates significant improvements in data retrieval speed and accuracy. The optimized data lake enables healthcare providers to access patient information more efficiently, leading to better support for clinical decision-making and enhanced patient care. The healthcare organization experiences reduced operational costs and improved overall efficiency, ultimately leading to better health outcomes for patients.

References

1. A. Bogatu, A. A. Fernandes, N. W. Paton, та N. Konstantinou, "Dataset discovery in data lakes," на 2020 IEEE 36th International Conference on Data Engineering (ICDE), IEEE, April 2020, с. 709-720.

2. C. Giebler, C. Gröger, E. Hoos, H. Schwarz, та B. Mitschang, "Leveraging the data lake: current state and challenges," на Big Data Analytics and Knowledge Discovery: 21st International Conference, DaWaK 2019, Linz, Austria, August 26–29, 2019, Proceedings 21, Springer International Publishing, 2019, с. 179-188.

3. N. Shakhovska, S. Fedushko, N. Melnykova, I. Shvorob, та Y. Syerov, "Big Data analysis in development of personalized medical system," Procedia Computer Science, т. 160, с. 229-234, 2019. https://doi.org/10.1016/j.procs.2019.09.461

4. F. Ravat, та Y. Zhao, "Data lakes: Trends and perspectives," на Database and Expert Systems Applications: 30th International Conference, DEXA 2019, Linz, Austria, August 26–29, 2019, Proceedings, Part I 30, Springer International Publishing, 2019, с. 304-313.

УДК 343.9, 004.8

Д.А. Клименко, *здобувач вищої освіти*
klimenkodarina12@gmail.com
Науковий керівник: Ю.В. Білявська, *доцент кафедри менеджменту, к. е. н., доцент*
y.biliavska@knute.edu.ua
Державний торговельно-економічний університет, Київ

ФІНАНСОВІ РИЗИКИ ТА ШАХРАЙСТВО: ЯК ШТУЧНИЙ ІНТЕЛЕКТ ДОПОМАГАЄ ЗАХИСТИТИ БАНКІВСЬКИЙ СЕКТОР

Фінансове шахрайство стало однією з головних загроз для бізнесу та банківського сектору, зокрема. Класичні способи виявлення аномалій та підозрілих транзакцій більше не гарантують належного рівня безпеки через зростаючу складність шахрайських схем. Штучний інтелект (ШІ) дозволяє аналізувати мільйони транзакцій у реальному часі, виявляючи закономірності, які неможливо розпізнати традиційними методами. В Україні ШІ-алгоритми активно застосовуються банками та фінансово-технічними компаніями для запобігання шахрайським операціям. У цьому дослідженні розглянемо реальні випадки впровадження штучного інтелекту у фінансовій безпеці та порівняємо ефективність різних методів.

ШІ-алгоритми застосовуються для виявлення фінансових шахрайств за допомогою різних методів машинного навчання та обробки даних. Основні підходи включають:

- Машинне навчання (ML): Використовує статистичні методи для побудови моделей, які можуть автоматично ідентифікувати аномальні транзакції. Наприклад, моделі класифікації та кластеризації допомагають розрізняти легальні та шахрайські операції.
- Глибинне навчання (Deep Learning): Нейронні мережі використовуються для аналізу великих масивів даних та виявлення прихованих закономірностей. Ці методи є особливо ефективними для розпізнавання складних схем шахрайства.
- Обробка природної мови (Natural Language Processing): Допомагає аналізувати текстові повідомлення та попереджувати фішингові атаки, а також виявляти загрози через автоматичний моніторинг спілкування.
- Виявлення аномалій (Anomaly Detection): Включає методи виявлення транзакцій, які відхиляються від нормального поведінкового патерну користувача. Цей підхід є корисним для виявлення нових типів шахрайства.

Представлені рівні впливу на загрозу віднесено до одного з трьох часових проміжків (рис. 1), залежно від того, коли прогнозується досягнення піку зрілості технології, що робить можливим її широке застосування в різних секторах економіки.

Вплив змін у цифрових технологіях	*2024-2025 рр.*	*2026-2029 рр.*	*2030-2032 рр.*
Високий	*Deep fake*, управління ризиками людського фактору, безпілотні транспортні засоби, поведінкова біометрія, технології віддаленої роботи, *Bring Your Own Identity* (*BYOI*), управління цифровим відбитком	Соціальний рейтинг, штучний інтелект із захистом персональних даних, диференціальна приватність, цифрові двійники, навчання роботі зі штучним інтелектом, проекційна телефонія	Потужний штучний інтелект (*AGI*), квантові комп'ютери, глобальний розум, штучний інтелект в окремих медичних напрямах, аугментація населення, двонаправлений нейрокомп'ютерний інтерфейс
Середній	Розумні бейджі, штучний інтелект, як бот, паспорт здоров'я, управління ідентифікацією пристроїв, професійна відеоаналітика	Генеративний штучний інтелект, односпрямований нейрокомп'ютерий інтерфейс, нейроморфна мікроелектроніка, мультихмарність	*ДНК* для зберігання інформації, *ДНК* гаджет, індивідуалізована медицина, цифрові близнюки, інтернет речей, децентралізовані автономні організації
Низький	*VR/AR*, 5G, технології соціальної дистанції та віддаленої роботи, блокчейн	Віртуалізація магазинів, створення віртуальних образів, помічників, партнерів	*XR*, 3D біодрук, біорозкладні сенсори, гіперперсоналізація, творчість *ШІ*
	Період часу		

Рис. 1. Зміна впливу перспективних цифрових технологій у фінансовому секторі економіки [5, с. 173]
Примітка Вісь X на графіку – час у роках, розділений на три проміжки: до 3 років, до 5 років, і більше 7 років. Вісь Y – вплив на ландшафт цифрових технологій, також розділена на 3 проміжки: низький, середній і високий.

Серед основних напрямків застосування ШІ у банкінгу – запобігання шахрайству та персоналізація послуг. Відомим кейсом є чат-бот зі штучним інтелектом Erica від Bank of America. З моменту запуску в 2018 році кількість взаємодій з цим чат-ботом перевищила 1,5 млрд. Він забезпечує цілодобову підтримку клієнтів, ефективно обробляючи запити та транзакції, що призводить до скорочення часу очікування та підвищення

рівня задоволеності клієнтів. Ще один приклад – банк Barclays, який використовує ШІ для виявлення шахрайства. Його система штучного інтелекту відстежує платіжні транзакції в режимі реального часу, виявляючи та запобігаючи потенційним шахрайським діям. Такий підхід не тільки захищає клієнтів, але й зміцнює їхню довіру до заходів безпеки банку.

U.S. Bank використовує штучний інтелект, щоб поліпшити якість обслуговування клієнтів, знизити ризики, боротися з шахрайством і відмиванням грошей, а також приймати рішення про видачу кредитів. Так, банк пропонує консультації за допомогою голосових помічників: Google Assistant, Siri від Apple і Alexa від Amazon. Технологія заснована на штучному інтелекті.

Банк збирає дані про клієнтів та їхніх транзакціях і за допомогою глибокого машинного навчання шукає закономірності, які допоможуть виявити недобросовісних клієнтів та скоротити кількість випадків шахрайства.

Організації, які впроваджують нові технології для управління фінансами, отримують набагато більші переваги, ніж спочатку розраховували: кількість помилок в роботі фінансових відділів знизилась в середньому на 37%; 72% організацій, що використовують технології штучного інтелекту повідомили, що у них з'явилося більш чітке уявлення про загальну ефективність бізнесу; 83% керівників вважають, що протягом п'яти років штучний інтелект дозволить зробити фінансове закриття повністю автоматичним; цифрові помічники підвищують продуктивність праці на 36% і дозволяють виконувати фінансовий аналіз на 38% швидше; кількість випадків викрадення акаунтів знизилася на 40%; час на блокування підозрілих операцій скоротився з кількох годин до кількох секунд.

Переважна більшість організацій вже освоїли перспективні технології. Так, українські банки, зокрема Sense Bank, також активно впроваджують штучний інтелект в банківські процеси. За словами Олександра Драгіна, керівника напряму з розвитку інформаційних технологій, Sense Bank, це допомагає збільшувати кількість автентифікованих клієнтів перед перемиканням на оператора, запобігати шахрайству та зменшувати середній час на обслуговування. Sense Bank використовує технологію біометрії (розпізнавання обличчя та голосу), а також метод моделювання, що дозволяє запобігти шахрайству операційно-соціальної інженерії.

Варто зазначити, що банківські установи можуть зіткнутися з серйозними викликами в майбутньому. Впровадження ШІ в банківській сфері ставить питання про забезпечення захисту конфіденційних даних клієнтів, одночасно використовуючи великі обсяги інформації для аналізу. У зв'язку з цим законодавчі обмеження можуть стати значною перешкодою для інтеграції нових ШІ-рішень. Не можна забувати, що багато банківських систем є застарілими, тому впровадження нових технологій потребує оновлення інфраструктури та інвестицій у навчання персоналу. Певним обмеженням також є неправильне визначення легальних транзакцій як шахрайських, адже це може привести до втрати клієнтів і зниження довіри до фінансових інститутів, саме тому необхідно постійно покращувати алгоритми для мінімізації таких помилок. Сюди ж віднесемо і етичні питання, бо використання ШІ у фінансовій безпеці може порушувати конфіденційність користувачів та створювати ризики неправильного використання даних.

Отже, використання штучного інтелекту в питаннях фінансової безпеки є одним із найбільш перспективних напрямів боротьби з шахрайствами в банківській сфері. Штучний інтелект дозволяє значно підвищити точність виявлення підозрілих транзакцій, автоматизувати процеси моніторингу та швидко реагувати на нові загрози. Порівняння ефективності ШІ та традиційних методів показало, що перший забезпечує вищу точність, швидкість обробки даних та здатність до самонавчання. Українські банки та фінансові компанії також активно інтегрують штучний інтелект у свої системи захисту, що дозволяє їм покращувати рівень безпеки та підвищувати довіру клієнтів. Подальші дослідження та вдосконалення систем штучного інтелекту сприятимуть більш ефективній протидії фінансовим загрозам.

Список літератури

1. Колесніков А. П., Карапетян О. М. Штучний інтелект: переваги та загрози використання. Efektyvna ekonomika. 2023. № 8. URL: https://doi.org/10.32702/2307-2105.2023.8.9

2. Як ШІ змінює українські банки - Fintech Insider. Fintech Insider - Дізнавайся першим. URL: https://fintechinsider.com.ua/yak-shi-zminyuye-ukrayinski-banky/

3. Теслюк С., Деркач О. Вплив фінтех-інновацій на банківський сектор україни. Наука і техніка сьогодні. 2023. № 8(22). URL: https://doi.org/10.52058/2786-6025-2023-8(22)-139-152.

4. Shestack Y., Biliavska Y., Osetskyi V., Mykytenko N. & Umantsiv Y. (2023). Devising a comprehensive method to manage digital competencies. Eastern-European Journal of Enterprise Technologies. № 3. P. 86–97.

5. Biliavskyi, V., Biliavska, Y., Umantsiv, Y., Shestack, Y., Zhurba, O. & Khavanov, A. (2024). Digital Technologies in the Financial Sector of the Economy. Financial and credit activity problems of theory and practice, 4 (57), 171–183.

УДК 629.7.02

Р.М. Минайленко, А.С. Безушко
aron70@ukr.net
Центральноукраїнський національний технічний університет, Кропивницький

АНАЛІЗ СУЧАСНОГО СТАНУ СИСТЕМ МОНІТОРИНГУ ПОБУТОВИХ ПРИСТРОЇВ РОЗУМНОГО БУДИНКУ

На теперішній час інтернет-технології все більше і більше входять в повсякденне життя людей як в промисловості так і в побуті. Прагнення людини до комфортного існування із мінімальними витратами часу на свої побутові проблеми призвело до виникнення різних ″розумних″ систем. Однієї з таких систем є розумний будинок.

Розумний будинок в сучасному розумінні стає об′єктом або зоною підвищеного комфорту в побуті людини, так як одна частина процесів в таких системах виконується автоматично, а іншою можна керувати віддалено з допомогою мобільних додатків встановлених в телефоні користувача. Такі системи швидко поширюються і стають більш доступними, оскільки дозволяють значно економити особистий час користувачів, зменшувати витрати на вирішення побутових проблем та підвищити рівень комфорту життя в таких приміщеннях.

Розумний будинок - це приміщення, де побутова техніка функціонально пов'язана між собою. Побутові пристрої можуть бути підключені до комп'ютерної мережі, що дозволяє проводити моніторинг та контролювати їх за допомогою комп'ютера (мікроконтролера) та забезпечувати віддалений доступ до них через мережу Інтернет. Завдяки використанню інформаційних технологій всі системи та пристрої розумного будинку узгоджують свої функції шляхом порівняння заздалегідь заданих програм і зовнішніх показників.

Особливістю розумного будинку є управління за допомогою пульта дистанційного керування, де людина може створити задану комфортну обстановку, натиснувши одну або кілька кнопок. Система сама аналізує зовнішні та внутрішні параметри приміщення і, керуючись результатами аналізу, виконує команди з певними параметрами, заданими користувачем.

На даний момент у компаній є багато інструментів для управління та моніторингу роботи побутових пристроїв, таких як розумні розетки, розумні лічильники тощо. Однак системного підходу до вирішення проблем, систем моніторингу немає, оскільки рішення про ввімкнення/вимкнення певного побутового приладу часто ініціюється користувачем вручну, а можливість автоматичної підтримки надійної роботи домашньої електромережі в разі високого навантаження часто відсутня.

Крім того, існуючі системи моніторингу побутових пристроїв розумного будинку часто вимагають додаткового мережевого обладнання, такого як шлюзи, маршрутизатори, а це призводить до додаткових вразливостей і знижує надійність систем.

Важливою складовою системи є сертифіковані протоколи обміну інформацією між апаратними елементами. Необхідно забезпечити збалансованість виробництва та споживання електроенергії, а для цього необхідно вміти контролювати стан споживачів та джерел енергії. Така система повинна бути бездротовою, захищеною від радіоперешкод і небажаного зовнішнього втручання, надійною та енергоефективною, мати невисоку вартість і можливість керування з комп'ютера (мікроконтролера) або мобільного телефону.

Актуальність теми дослідження пов′язана з тим, що компанії, які займаються розробкою і впровадженням розумних будинків на даний час не виробили єдиної методології щодо їх створення і експлуатації. Такі системи мають бути економічними, ергономічними, надійними та захищеними, тому відкриваються великі можливості для їх вдосконалення.

УДК 004.005

Я. І. Линник[1, 2], С.В. Науменко[1,3]
lynnyk265@gmail.com, igarasym@space-it.com.ua
[1]*Черкаський національний університет імені Богдана Хмельницького, Черкаси*
[2]*Студент 4 курсу*
[3]*Науковий керівник*

ПЛАТФОРМА КЕРУВАННЯ ОНЛАЙН-ОГОЛОШЕННЯМИ

У сучасному цифровому середовищі платформи для розміщення онлайн-оголошень (OLX, Shafa) користуються великою популярністю серед широкого кола користувачів. З огляду на високі навантаження та необхідність забезпечення безперервного доступу, ефективність таких систем залежить від правильного проєктування їхньої архітектури.

В основі реалізації платформи лежить мікросервісна архітектура, що дозволяє гнучко масштабувати окремі компоненти системи відповідно до їх навантаження. Це суттєво відрізняється від монолітного підходу, де при зростанні навантаження доводиться масштабувати увесь застосунок, навіть якщо навантаження виникає лише на одну його частину. Мікросервісна архітектура забезпечує кращу стійкість до збоїв, незалежне розгортання та оновлення кожного модуля, а також підвищує загальну гнучкість системи.

Одна з ключових особливостей онлайн-платформ такого типу — нерівномірне розподілення навантаження між сервісами. Найчастіше найбільше навантаження отримує пошуковий сервіс, оскільки кожен користувач взаємодіє з системою саме через пошук оголошень. У традиційній монолітній архітектурі збільшення продуктивності вимагає масштабування всього застосунку, навіть якщо перевантажено лише одну його частину.

Для обміну повідомленнями між сервісами використовується Apache Kafka [1], яка відіграє роль асинхронного брокера повідомлень. Це дозволяє досягти високої швидкодії та надійності передачі даних між компонентами системи, зменшити зв'язаність сервісів і забезпечити обробку подій у реальному часі.

Пошукова функціональність реалізована за допомогою Elasticsearch [2] — потужного інструмента для повнотекстового пошуку. Це дає змогу користувачам швидко знаходити необхідні оголошення за ключовими словами, категоріями, місцезнаходженням, ціною та іншими параметрами.

Основні ключові мікросервіси платформи керування онлайн-оголошеннями:

Catalogue-service — основний мікросервіс для роботи з оголошеннями. Забезпечує повний CRUD-функціонал для оголошень. Підтримує створення оголошення, оновлення інформації оголошення, збереження каталогу з повним набором інформації про оголошення.

User-service —мікросервіс, що відповідає за обробку всієї інформації про користувачів: реєстрацію, вхід, відновлення пароля, керування особистими даними. Використовує JWT (JSON Web Token) для реалізації механізмів авторизації та автентифікації.

Order-service — мікросервіс, що служить для обробки операцій, пов'язаних із купівлею-продажем товарів. Фіксує історію замовлень, їхній поточний статус, зберігає інформацію про покупців і продавців. Може бути розширений під інтеграцію з платіжними системами, наприклад, LiqPay чи Stripe.

Search-service — мікросервіс, що реалізує пошукову систему за допомогою Elasticsearch — швидкого пошукового рушія. Забезпечує повнотекстовий пошук за заголовком, описом, категоріями, регіоном, ціною, а також підтримує сортування та фільтрацію результатів. Пошук оптимізований під високу кількість одночасних запитів, з можливістю масштабування для покриття пікового навантаження.

Chat-service — мікросевріс, що дає змогу користувачам переписуватись в межах платформи. Підтримує обмін текстовими повідомленнями в реальному часі за допомогою WebSocket. Повідомлення зберігаються у базі даних з можливістю подальшої модерації або архівування. Можливе розширення до підтримки вкладень (зображення, документи).

Список літератури

1. Офіційна сторінка документації Apache Kafka [Електронний ресурс] – Режим доступу: https://kafka.apache.org/documentation

2. Офіційна сторінка документації Elasticsearch [Електронний ресурс] – Режим доступу: https://www.elastic.co/docs

УДК 629.7.02

Р.М. Минайленко, Д.А. Закарлюка, А.С. Коваленко
aron70@ukr.net
Центральноукраїнський національний технічний університет, Кропивницький

АНАЛІЗ МЕТОДІВ ДОСЛІДЖЕНЬ ПРОЦЕСІВ В МЕРЕЖІ ПЕРЕДАЧІ ДАНИХ

На сьогоднішній день майже не залишилося персональних комп'ютерів не підключених до мережі зв'язку. Це сприяє на постійний ріст попиту на послуги зв'язку та інформацію. Якість послуг мережі передачі даних характеризується показниками, що оцінюють властивості послуг, які задовольняють вимоги клієнтів щодо швидкості, точності передачі повідомлень і надійності засобів зв'язку, а також відображають процес обслуговування споживачів, тобто доступність, зручність користування засобами зв'язку, надійність і стійкість зв'язку, тощо.

Потреба модернізації мережі передачі даних, а саме збільшення швидкості передачі даних є однією з найпоширеніших причин модернізації мережі. Якщо продуктивність мережі зв'язку недостатня, першим, що треба зробити – визначити рівень завантаженості каналів. Зазвичай вважається, що ємність каналу слід збільшити, коли рівень його завантаженості досягає 70%. Якщо пропускна здатність каналу достатня, то причина може бути в продуктивності обладнання. Іншою причиною низької продуктивності може бути пов'язана з тривалою затримкою у мережі. Причиною затримки може бути повільна обробка повідомлень.

Для дослідження мережі передачі даних можуть застосовуватися такі методи як аналітичний, математичній, метод фізичного експерименту та імітаційне моделювання.

Аналітичний метод – це метод, при якому здійснюється логічний аналіз будь-якої прогнозованої ситуації. На основі розробленої моделі експерти розробляють логіку розвитку об'єкту, встановлюють умови його розвитку у майбутньому, дають кількісні оцінки важливості можливих шляхів розвитку. Проте складність процесів моделі мережі передачі даних робить неможливим точний аналітичний аналіз одночасних процесів у мережі. Оскільки такі процеси як надходження у мережу запитів на передачу повідомлень від одного абонента мережі до іншого, відмови та відновлення елементів мережі та розмір повідомлень мають стохастичний характер, тому це робить неможливим використання аналітичного методу.

Математичний метод дослідження являє собою способи кількісного та структурного вивчення явищ, у процесі якого їх властивості й закономірності визначаються за допомогою формальних моделей, кількісних характеристик, функціональних відношень. Недоліком даного методу є складність обчислень точності отриманих результатів, тому даний метод не може бути застосованим для дослідження мережі передачі даних .

Ще один можливий варіант дослідження моделі – фізичний експеримент. Даний метод базується на побудові лабораторної фізичної моделі системи у зменшених масштабах і проведені експериментів на даній моделі, тому не завжди є доречним через можливість великих затрат на побудову.

Інший метод дослідження інформаційних систем – імітаційне моделювання. Новітні технології моделювання значно спростили та прискорили процес побудови та дослідження моделі. Сучасні технології моделювання можуть представляти результати моделювання засобами 3D графіки й анімації, що значно спрощує сприйняття імітаційної моделі та допомагає знайти спільну мову між спеціалістами з моделювання систем та спеціалістами, що працюють у галузі, яка моделюється.

Перевага даного методу полягає у можливості змін процесу подій досліджуваної системи з реальних часових одиниць на прискорений процес зміни подій у темпі роботи програми. Ефективність імітаційного моделювання може оцінюватися точністю і правильністю результатів моделювання, часом побудови та роботи з імітаційною моделлю, мінімальними витратами машинних ресурсів, такі як час і пам'ять, а також вартістю розробки й експлуатації моделі.

Отже випадкові величини такі як частота надходження повідомлень у систему, розміри повідомлень, а також збої, які виникають під час роботи каналів передачі даних, передбачити математичним, аналітичним та фізичним методами неможливо, тому найкращим методом для досліджень стохастичних моделей є метод імітаційного моделювання.

УДК 004.056, 004.75

М.Р. Безкоровайний, *к.е.н., доцент* Н. Л. Чернова,
mykyta.bezkorovainyi@cs.khpi.edu.ua
Національний технічний університет «Харківський політехнічний інститут», Харків

ІНТЕЛЕКТУАЛЬНІ АСИСТЕНТИ В КУЛІНАРІЇ: АНАЛІЗ ФУНКЦІОНАЛУ ТА СУЧАСНИХ ТРЕНДІВ

Розглянуто сучасний ринок інтелектуальних асистентів, відокремлено особливості функціоналу, що притаманні саме кулінарній галузі. Визначено входи і виходи для відповідної моделі чорного ящика.

Глобальний ринок інтелектуальних асистентів є одним з тих, що демонструють найшвидші темпи зростання. Так, за оцінками експертів його об'єм на кінець 2023 року вже становив приблизно 3,24 блн. дол. США, а за період до 2032 року сукупний середньорічний темп зростання прогнозується на рівні 22-24%[1-2]. Інтелектуальний асистент (ІА) – це інтерактивна програма, що призначена для надання допомоги користувачам у вирішенні різноманітних завдань, таких як надання інформації, автоматизація рутинних процесів, персональна допомога, голосова взаємодія, інтеграція з іншими програмами, створення унікального контент. Базові кроки, що реалізує у своїй роботі ІА, це отримання повідомлення від користувача, обробка повідомлення, отримання інформації та генерація відповіді. Галузі, що ефективно впроваджують в свої процеси ІА, це роздрібна торгівля, фінанси, автомобільна промисловість, побутова електроніка, охорона здоров'я, туризм, державне управління та кулінарія[3-4]. Кулінарні ІА керують процесами виробництва та збереження продуктів харчування, генерують рецепти страв, створюють меню, планують дієти. Станом на початок 2025р. популярними ІА є: Chef Watson, ChefGPT, Dishgen, Let's Foodie, FoodAI, SuperCook, PlantJammer, EmptyMyFridge, Cooklist[5-6].

Кінцевими користувачами зазначених ІА можуть виступати як окремі індивідууми, так і бізнес-структури, як виробничі, так і сервісні компанії. Аналізуючи ІА за допомогою моделі чорного ящика, вхідними параметрами є перелік наявних/бажаних продуктів харчування та їх об'єми/кількість; калорійність страви; кількість персон, для яких потрібно приготувати страву; загальний час приготування; наявні кухонні пристрої; припустимість заміни інгредієнтів; особливості дієти. Зазначені параметри можуть відігравати роль як керованих так і нормативних. На виході отримуємо рецепт та/або інструкцію по приготуванню або окремої страви, або набір рецептів, якщо вирішувалась більш складна задача формування меню або дієти. Якщо ІА є вбудованим компонентом більш складної системи (наприклад, системи смарт-дім), на виході очікуємо готову страву.

ІА асистенти загалом можуть мати досить різні архітектури в залежності від свого цільового призначення. Кулінарні асистенти як правило мають гібридну архітектуру, яка поєднує в собі переваги реактивних архітектур та архітектур, заснованих на знаннях. Перший варіант використовується для підтримки процесів реалізації рутинних задач (керування процесом приготування за шаблонним рецептом). Впровадження другого варіанту дозволяє значно розширити функціональні можливості кулінарного ІА та вирішувати складні погано структуровані задачі. У більшості проаналізованих кулінарних ІА основною є функція генерації рецептів, яка реалізована за допомогою методів машинного навчання та штучного інтелекту. В залежності від набору вхідних параметрів, ІА реалізують додатковий функціонал персоналізації рецепту, дозволяючи враховувати різноманітні обмеження. Саме ступень цієї персоналізації є одним з ключових факторів, за яким відрізняються кулінарні асистенти.

Список літератури

1. Business research insights. 17 March 2025, URL: https://www.businessresearchinsights.com/market-reports/virtual-assistant-market-111910 ;
2. Polaris market research. Jan-2024, URL: https://www.polarismarketresearch.com/industry-analysis/intelligent-virtual-assistant-market ;
3. Fortune business insights. https://www.fortunebusinessinsights.com/industry-reports/intelligent-virtual-assistant-market-100758 ;
4. Imarc. https://www.imarcgroup.com/intelligent-virtual-assistant-market ;
5. Jitendra Jain. Inventcolabs. December 23, 2024 https://www.inventcolabssoftware.com/blog/ai-recipe-generator-app/ ;
6. Fritz. 6 Best AI Recipe Generators in 2024: Our Top Picks. Feb 17, 2024 https://fritz.ai/best-ai-recipe-generators/ .

УДК 621.317

В.І. Солодка, М.О. Патлаєнко, Т.В. Урсу, О.А. Соломатін
valyaonas@gmail.com, nick_msa@ukr.net, tanaursu2000@gmail.com, oleg1greenberg@gmail.com,
Державний університет інтелектуальних технологій та зв'язку , Одеса

ОЦІНКА МЕТОДІВ ВИМІРЮВАННЯ ЗАВАД НА ПАРАМЕТРИЧНУ СТАБІЛЬНІСТЬ РАДІО СИГНАЛІВ

Вимірювання параметрів радіотехнічні систем – це ланцюжок дослідно-розрахункових дій, які виконуються для визначення величини, що описує конкретний предмет або процес. Сьогодні радіотехнічні системи розвиваються на базі вбудованих обчислювальних систем та завдяки впровадженню передових інформаційних технологій.

Най-важливіша та основна вимога до будь-якої радіотехнічної системи полягає в забезпеченні надійного та своєчасного отримання інформації з обмеженого енергетичного випромінювання. Надійному прийому інформації в радіотехнічних системах перешкоджає низка факторів.

Найчастіше реальний сигнал неможливо спостерігати безпосередньо, а замість цього маємо справу з сигналом, спотвореним шумом, або його приблизним розрахунком. При цьому існує величезна кількість різних різновидів шуму.

Шуми [1] становлять серйозну проблему в науці інженерії та метрології, адже вони задають границі точності вимірювань та чутливості електронних систем до сигналів. Шум різниться за частотним спектром, який визначає його "колір" [2].

Для дослідження впливу стохастичних завад різної природи на параметричну стабільність радіо сигналів будемо використовувати вісім різних завад:

- білий шум (завади) це шум який є постійним і його спектральні складові рівномірно розподілені по всьому діапазону частот при використані у радіотехнічних системах [3].

- рожевий шум – це є біле світло, який охоплює весь спектр кольорів, а червоне світло має низькі частоти. Так само у техниці рожевий шум – це шум, що містить менше високих або верхніх частот.

- коричневий шум – один з багатьох кольорів шуму, до яких також відносяться білий, рожевий і синій. Коричневий шум має ще більш виражений зсув енергії у бік низького спектру.

- високочастотним білим шумом, щей ого називають синім або блакитний шум, який часом іменують, є кольором шуму зі спектральною щільністю, що прямо пропорційна його частоті. Тобто потужність та енергія сигналу збільшуються зі зростанням частоти. У звукових програмах блакитний шум використовують для дизерингу – процесу, при якому шум додається до доріжки з метою згладжування звуку та зменшення гучності спотворень.

- фіолетовий шум має зростання 6 дБ на октаву, що робить його своєрідним диференційованим сигналом шуму з характерною траєкторією. Завдяки цій властивості, він може бути корисним для фільтрації певних високочастотних звуків, що супроводжують, скажімо, тинітус або деякі форми втрати слуху.

- імпульсний – це шум, який складається з одного або кількох звукових імпульсів, які різко зростають і видко спадають, причому тривалість кожного з них менше ніж 1с. Шкідливість шуму залежить від декількох факторів: тривалості впливу шуму, інтенсивності, частотного спектра і того, як він змінюється в часі.

- гаусів шум, або це називають випадковий гаусів процес, виникає при підсумовуванні статистично незалежних білих шумів.

- рівномірний шуми це модель описуюча адитивний шум, розподіл якого є рівномірним. Він часто зустрічається у дослідженнях на зображеннях після квантування або ж тих, що зазнали спотворень під час передавання.

Нижче подано частину коду, написаного мовою програмування Matlab, де показано суть розробленого методу.

```
%% Параметри сигналу
Fs = 10000;  % Частота дискретизації (Гц)
T = 1;       % Тривалість сигналу (сек)
t = 0:1/Fs:T-1/Fs;  % Часова вісь
A = 1;   % Амплітуда сигналу
f = 1;  % Частота сигналу (Гц)
phi = 0; % Фаза (рад)
%% Сигнали з шумами
signals_with_noise = {white_noise, pink_noise, brown_noise, blue_noise, violet_noise, ...
                      impulse_noise, gaussian_noise, uniform_noise};
noise_names = {'Білий', 'Рожевий', 'Коричневий', 'Синій', 'Фіолетовий', ...
```

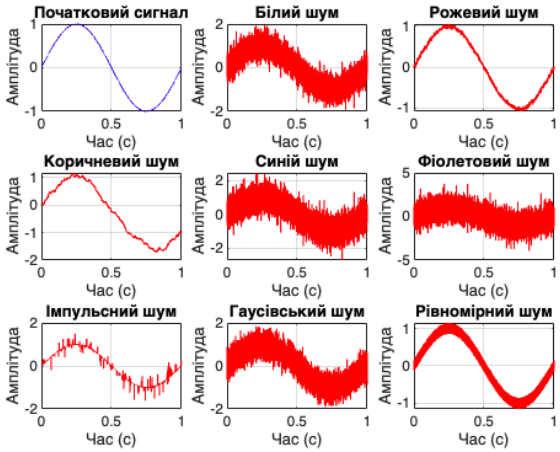


Рис. 1 – Генерація шумів при $\Delta f = 1$ Гц , $\Delta U =$ 1В , $\Delta \overline{|\varphi,|}^{c} = 1$

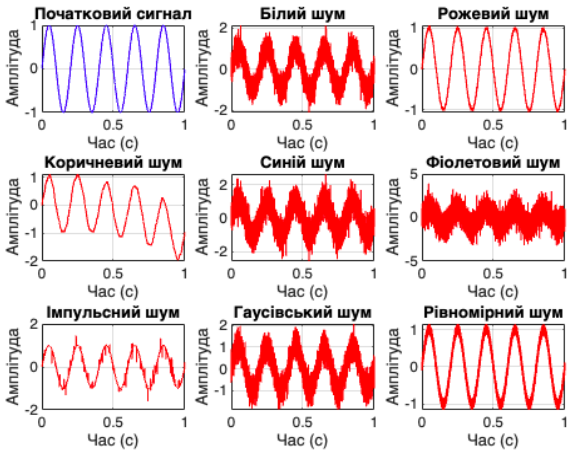


Рис. 2 – Генерація шумів при $\Delta f = 10$ Гц , $\Delta U =$ 1В , $\Delta \overline{|\varphi,|}^{c} = 1$

Таблиця 1

Оцінка похибок сигналу з різними шумами для параметрів з рис.1

Шум	ПОХИБКИ			
	Δf,Гц	ΔU, В	$\Delta \overline{\|\varphi,\|}^{c}$	Δ *Form*
Білий	0.0000	0.0058	0.0013	0.0793
Рожевий	0.0000	0.0064	0.0006	0.0006
Коричневий	0.0000	0.3465	0.1250	0.0305
Синій	0.0000	0.0000	0.0000	0.1444
Фіолетовий	0.0000	0.0000	0.0000	0.3099
Імпульсний	0.0000	0.0003	0.0006	0.0010
Гаусів	0.0000	0.0010	0.0057	0.0801
Рівномірний	0.0000	0.0007	0.0000	0.0074

Таблиця 2

Оцінка похибок сигналу з різними шумами для параметрів з рис.2

Шум	ПОХИБКИ			
	Δf,Гц	ΔU, В	$\Delta \overline{\|\varphi,\|}^{c}$	Δ *Form*
Білий	0.0000	0.0043	0.0000	0.0795
Рожевий	0.0000	0.0025	0.0000	0.0006
Коричневий	0.0000	0.0302	0.0013	0.0921
Синій	0.0000	0.0000	0.0000	0.1444
Фіолетовий	0.0000	0.0000	0.0000	0.3099
Імпульсний	0.0000	0.0007	0.0000	0.0010
Гаусів	0.0000	0.0056	0.0013	0.0807
Рівномірний	0.0000	0.0006	0.0000	0.0075

Для дослідження теми необхідно зробити ґенерацію шумів. Фундаментальним завданням вимірювання шуму та дослідження даної теми є спостереження за паремтрами сигналу, а саме амплитудою, частотою та фазою при генерації шумів.

Результат самого дослідження показанно на рис. 1 та рис. 2, оцінки похибок сигналу параметрів показані у таблицях 1 та 2. Результати дослідження показали, що різнокольорові шуми не впливають на оцінку похибки частоту самого сигналу.

Список літератури

1. Dunn J. E. Detection and classification of frequency-hopped spread spectrum signals: Ph.D. Electrical Engineering and Computer Engineering. Iowa State University, 1991. 134 p.

2. Chevva L., Sagar G. V. R. FH Signal Interception Based on the Time-Frequency Spectrogram by Image Enhancement Techniques // International Journal of Engineering Research and Applications, 2012. Vol. 2, Issue 2. P. 687−692.

3 Класифікація сигналів. [Електронний ресурс] Режим доступу: https://studme.org/285792/tehnika/

УДК 004.8

П.С. Усік, *старший викладач, доктор філософії(PhD)*
usikps@kntu.kr.ua
Центральноукраїнський національний технічний університет, Кропивницький

ОПЕРАЦІЙНІ СИСТЕМИ НА БАЗІ ШТУЧНОГО ІНТЕЛЕКТУ: НОВИЙ ЕТАП ЕВОЛЮЦІЇ ЦИФРОВОГО СЕРЕДОВИЩА

У сучасних умовах розвитку цифрових технологій виникає потреба у принципово нових підходах до організації обчислювальних середовищ. Операційна система, заснована на штучному інтелекті (ОС ШІ), — це не просто програмне забезпечення для управління апаратними ресурсами, а інтелектуальна платформа, що інтегрує машинне навчання (ML), обробку природної мови (NLP) та комп'ютерне бачення для розширення і поглиблення своїх базових функцій. Така система покликана зробити пристрої розумнішими, адаптивнішими та здатними до самонавчання на основі взаємодії з користувачем. На відміну від традиційних ОС, які працюють за фіксованими алгоритмами, ОС ШІ може прогнозувати потреби, автоматизувати рутинні процеси та ухвалювати рішення в режимі реального часу, трансформуючи платформу з пасивної в активного цифрового асистента.

Операційні системи зі штучним інтелектом можуть стати ефективним підходом до побудови сучасних обчислювальних середовищ, особливо з огляду на зростаючу складність технологій та потребу в адаптивних системах. Такі ОС поєднують класичні принципи керування ресурсами з можливостями машинного навчання та обробки природної мови. Ось які переваги дає використання штучного інтелекту в операційних системах:

1. Оптимізоване управління ресурсами. На відміну від традиційних ОС, які працюють за жорстко заданими алгоритмами, ШІ-системи використовують машинне навчання для аналізу та прогнозування навантаження. Це дозволяє динамічно розподіляти ресурси: оперативну пам'ять, процесорний час, енергоспоживання.

2. Адаптація до користувача. ШІ дозволяє ОС враховувати індивідуальні звички користувача. Система вивчає часто використовувані додатки, реакції на повідомлення, типові дії та може автоматично налаштовувати середовище. Впровадження обробки природної мови (NLP) і голосових інтерфейсів дозволяє виконувати дії за допомогою звичайних команд, спрощуючи роботу користувачів.

3. Проактивна безпека. ШІ-алгоритми постійно моніторять поведінку системи, аналізують мережевий трафік, дії користувача та програм. Це дозволяє виявляти підозрілі активності ще до появи загроз. Такі ОС здатні не лише виявити потенційну атаку, але й автоматично ізолювати уразливі компоненти, змінити налаштування безпеки або обмежити доступ, запобігаючи поширенню шкідливого ПЗ.

4. Підвищена ефективність і масштабованість. ОС із ШІ можуть гнучко працювати на різних апаратних платформах — від IoT-пристроїв до хмарних дата-центрів. Вони аналізують навантаження, передбачають використання даних і адаптують операції з пам'яттю, зберіганням та введенням/виведенням інформації. Це особливо важливо для систем з непередбачуваним навантаженням.

5. Самонавчання та розвиток. ОС зі ШІ здатні постійно вдосконалювати власні алгоритми. Вони адаптуються до змін у середовищі та поведінці користувача без втручання адміністратора. Сучасні системи також можуть аналізувати контекст, надавати рекомендації щодо покращення продуктивності, вирішувати проблеми або автоматизувати повторювані завдання.

Попри численні переваги, операційні системи зі штучним інтелектом мають і низку суттєвих недоліків, що потребують серйозного осмислення. По-перше, це питання конфіденційності та безпеки: обробка великих обсягів персональних даних підвищує ризик витоків інформації. По-друге, надмірна довіра до рішень ШІ може призвести до втрати критичного мислення у користувачів. Також зростає занепокоєння щодо можливого витіснення робочих місць через автоматизацію, а самі системи можуть успадковувати упередження з навчальних даних, що створює ризики несправедливих рішень. Нарешті, складність внутрішніх алгоритмів ускладнює діагностику помилок і зрозумілість рішень, що особливо критично для відповідальних застосувань.

Таким чином, інтеграція штучного інтелекту в операційні системи відкриває нові горизонти для розвитку обчислювальних технологій. ОС зі ШІ демонструють значний потенціал у підвищенні ефективності, гнучкості, безпеки та інтуїтивності взаємодії з користувачем. Водночас повноцінне впровадження таких систем вимагає уважного врахування супутніх ризиків — від етичних і соціальних до технічних і безпекових. Розробка й експлуатація ОС на основі ШІ має базуватись на принципах прозорості, відповідальності та дотримання конфіденційності, щоб забезпечити їх безпечне та ефективне функціонування в сучасному цифровому середовищі.

УДК 004.331.4.614.8

К.М. Марченко, І.В. Варченко, К.Є. Крюков
k_marchenko@i.ua
Центральноукраїнський національний технічний університет, Кропивницький

СТВОРЕННЯ БЕЗПЕЧНИХ УМОВ ПРАЦІ ІТ-ПРАЦІВНИКІВ З УРАХУВАННЯМ ЕЛЕКТРОМАГНІТНОГО ВИПРОМІНЮВАННЯ

В умовах інформаційного суспільства стрімке зростання обсягів інформації, що використовується, зумовлює відповідне збільшення як потужності технічних засобів, що передають та оброблюють інформацію, так і щільності їх розповсюдження. Робота засобів передавання та обробки інформації супроводжується, зокрема, електромагнітним випромінюванням (ЕМВ), тому працівники ІТ-сфери крім дії ЕМВ від електричних мереж та електроустановок, підлягають також додатковій дії ЕМВ, що утворюються персональними комп'ютерами, мережевим та серверним обладнанням, перефірійними пристроями тощо.

Ступінь опромінення працівників залежить від кількості джерел випромінювання, їхньої потужності, тривалості впливу, екранування, розміщення джерел усередині та зовні приміщень. Наприклад, Wi-Fi роутер є джерелом ЕМВ потужністю 45-65 мкВт/см^2, потужність ЕМВ в приміщенні від вишки мобільного зв'язку 10-60 мкВт/см^2, смартфона в режимі передавання даних в напрямку мобільної станції - 80-100 мкВт/см^2. За результатами досліджень як вчених, так і фахівців-практиків [1, 2] постійний вплив ЕМВ на людину викликає такі захворювання та розлади як втрата концентрації уваги, втома, головний біль, порушення роботи окремих органів – шлунку, печінки, підшлункової залози. Погіршується діяльність серцево-судинної системи, спостерігаються зміни показників білкового та вуглеводневого обміну, змінюється склад крові, зафіксовані зміни на рівні клітин. а в довгостроковій перспективі ЕМВ навіть може привести до розвитку ракових пухлин.

Дія ЕМВ на працівників та споживачів регламентується державними санітарними нормами [3], але зміни до цих стандартів, які відбулися в останні роки [4] привели до збільшення гранично-допустимої дії ЕМВ спочатку з 2,5 мкВт/см^2 до 10 мкВт/см^2, а потім і до 100 мкВт/см^2, що суттєво знижує захищеність працівників від негативного впливу ЕМВ. Проблемою також є низький рівень контролю дотримання цих норм в Україні, що також знижує захищенність працівників.

Таким чином, працівники, зайняті у сфері інформаційних технологій, піддаються надлишковому впливу електромагнітного випромінювання, що викликає необхідність додаткових досліджень у напрямку створення безпечних умов праці з урахуванням цього чинника, розробки відповідних заходів та засобів безпеки. Для отримання даних про реальний рівень ЕМВ в офісних та виробничих приміщеннях пропонується систематизовані вимірювання рівню ЕМВ з заданим невеликим кроком за площею та складання карти густини електромагнітного поля. Для вимірювання рівню ЕМВ в теперішній час існують даступні прилади, наприклад, детектор електромагнітного випромінювання BENETECH GM3120 [5]

Висновок. Отримання карти густини поля ЕМВ для певного приміщення дозволить:

1. Отримати реальні дані за дією ЕМВ на працюючих:
2. Атестувати приміщення та робочі місця за виконанням норм ЕМВ;
3. Оптимізувати відносне розміщення робочих місць та обладнання, яке є джнонлами ЕМВ;
4. За необхідністю вибрати та установити додаткові засоби захисту від надлишкового ЕМВ.

Список літератури

1. Електромагнітне випромінювання. Частина 1. URL: https://opcb.kpi.ua/?p=3936

2. Чорний О. П., Никифоров В. В., Родькін Д. Й., Ноженко В. Ю. Сучасний стан досліджень впливу електромагнітних випромінювань на організм людини. // Інженерні та освітні технології в електротехнічних і комп'ютерних системах. № 2/2013 (2), 2013. - с. 112-124.
URL: https://eetecs.kdu.edu.ua/2013_02/EETECS2013_0208.pdf

3. ДСН 239-96. Державні санітарні норми і правила захисту населення від впливу електромагнітних випромінювань. URL: https://zakon.rada.gov.ua/laws/show/z0488-96#Text

4. Про затвердження Зміни до Державних санітарних норм і правил захисту населення від впливу електромагнітних випромінювань» від 30.11.2020 р. № 2760.
URL: https://zakon.rada.gov.ua/laws/show/z0026-21#Text

5. Детектор электромагнитного излучения BENETECH GM3120.
URL: https://protester.com.ua/gm3120/?srsltid=AfmBOoo9O7XsfYF-DMEjwyiRiTsl59ZLrxzz-MNWj-7dKzE8pI74IEXY

УДК 378:004.384

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

СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ DATA MINING В УКРАЇНІ

У сучасному інформаційному суспільстві зростає попит на методи, що дозволяють ефективно аналізувати великі обсяги даних. Data Mining (інтелектуальний аналіз даних) є одним з ключових елементів цифрової трансформації, що забезпечує ухвалення обґрунтованих рішень у багатьох сферах. В Україні ця галузь активно розвивається, хоча й залишається певний розрив із провідними країнами світу.

Data Mining — це процес виявлення неочевидних, раніше невідомих, але потенційно корисних закономірностей у даних. Його цілі включають: класифікацію (наприклад, розпізнавання шахрайства у фінансах); кластеризацію (групування споживачів за поведінкою); прогнозування (наприклад, прогноз попиту або поведінки клієнтів); виявлення аномалій (у кібербезпеці або охороні здоров'я).

Сучасний стан розвитку Data Mining в Україні характеризується такими напрямками розвитку:

1. Наука та освіта.

Провідні ВНЗ України активно включають курси з Data Science, Data Mining, Machine Learning у навчальні програми, в тому числі і Центральноукраїнський національний технічний університет, зокрема викладачі кафедри кібербезпеки та програмного забезпечення. Серед прикладів можна назвати також дослідження, реалізовані науковцями КПІ, які створили модель для передбачення трафіку в Києві [1].

2. Бізнес-середовище.

Компанії Grammarly, Reface, YouScan, DataRoot Labs активно впроваджують механізми Data Mining в своїй діяльності, як під час створення і реалізаціїї конкретних проектів так і для аналізу і покращення внутрішніх бізнес-процесів. Також один із ведучих гравців на фінансовому ринку країни, а саме Monobank використовує аналітику Data Mining для персоналізації банківських послуг.

3. Державний сектор.

Портал data.gov.ua фіційний державний портал відкритих даних України, створений для забезпечення прозорості, підзвітності державних органів, а також для стимулювання розвитку інновацій та стартапів на основі відкритої інформації використовується для аналітики та створення сервісів. Так на порталі можна знайти такі реальні кейси як Opendatabot — платформа, що використовує дані з реєстрів юридичних осіб, судових рішень, податкових повідомлень, YouControl — моніторинг ризиків контрагентів, візуалізація зв'язків між компаніями.

Проблеми та виклики з якими на сьогоднішній день стикається подальший розвиток аналітичних систем це по-перше нерівномірний розвиток Data Mining у різних секторах економіки та суспільства, необізнаність більшості громадян щодо умов та можливостей використання аналітичних систем. По-друге це відтік кваліфікованих кадрів за кордон. Подібні процеси відбуваються в усіх секторах економіки, з огляду на сучасний стан політичної та економічної ситуації в державі. Однак втрата кваліфікованих фахівців аналітиків призводить до відчутних наслідків, зокрема через складність і тривалість їх підготовки. Недостатня інфраструктура для зберігання та обробки даних також впливає на загальний стан галузі оскільки обробка та аналіз великих об'ємів даних напряму пов'язане із вільним та швидким доступом до них, надійних каналах передачі даних та захищених сховищах інформації. Одним із вагомих недоліків є також проблеми з доступом до структурованих даних у держсекторі, хоча це один із найважливіших напрямків використання інструментів Data Mining через свою гнучкість, прозорість, точність та ефективність.

Серед перспектив розвитку Data Mining в Україні потрібно відмітити науково-технічні, а саме розвиток AI/ML інфраструктури при вищих навчальних закладах та наукових центрах а також хмарні рішення (AWS, Azure) для аналізу даних. Існують також і економічні напрямки, це зростання інвестицій у data-driven бізнес та потенціал у AgTech, телемедицині, логістиці. У сфері державної підтримки: програми "Дія.Освіта", "Дія.Сіті" також сприяють розвитку інновацій.

Розвиток Data Mining в Україні має позитивну динаміку. Поєднання академічної підготовки, практичного застосування в бізнесі та ініціатив у держсекторі формує підґрунтя для сталого розвитку галузі.

Список літератури

1. *Lysenko I.A.* Information technology of developing test kits based on software requirements / I.A. Lysenko, O.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Delhi (India), 2016. – Volume 6, Issue 1. – P. 35-38.

УДК 004.8 656.02

О.А. Кислун., *доцент*, Ю.М. Пархоменко, *доцент*
kyslun@gmail.com, parhomenkoym@ukr.net
Центральноукраїнський національний технічний університет, Кропивницький

ОГЛЯД МОЖЛИВОСТЕЙ КЛАСИЧНИХ МЕДІВ ПОШУКУ ОПТИМАЛЬНИХ МАРШРУТІВ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ ВИРОБНИЧОЇ ЛОГІСТИКИ

Сучасне виробництво, як процес, може бути ефективним лише при оптимальному управлінні матеріальними та інформаційними потоками, за що й відповідає виробнича логістика. Підвищення ефективності виробництва при забезпеченні якості та його гнучкості за рахунок такої складової, як логістика, можливе за рахунок мінімізації витрат постачання при своєчасності.

В основу витрат постачання покладені витрати доставки (витрати на просте переміщення, що зазвичай прямопропорційні довжинам переміщень - шляхам).

Методика пошуку оптимальних маршрутів за мінімізаційним критерієм за попередні роки досить добре розроблювалась та на даний час представлена досить широким рядом напрацювань у вигляді алгоритмів пошуку мінімальних маршрутів, які вважають класичними.

Існуючі алгоритми пошуку найкоротшого шляху: Дейкстри, Форда, Беллмана-Форда, Флойда-Воршелла, пошуку A*, пошуку D*, Джонсона та інші.

Аналіз класичних алгоритмів показав, що незалежно від типу задач, що вони розв'язують (пошук найкоротшого шляху з одного входу, пошук найкоротших шляхів з одним виходом чи пошук найкоротшого шляху між парою пунктів), вони можуть ефективно використовуватись для мінімізації витрат на переміщення безпосередньо, проте вони не враховують супутні витрати при використанні проміжних пунктів маршрутів, але ж витрати на вантажно-розвантажні роботи та складські витрати на збереження (загалом дані витрати в свою чергу враховують термін та умови зберігання) можуть суттєво впливати на загальні логістичні витрати, а відповідно з позиції повних наявних витрат мінімальність може й не досягатися.

Отже, враховуючи зазначене, наявні класичні алгоритми для пошуку оптимальних маршрутів використати безпосередньо неможливо, проте ввівши додаткові ланки до опису маршрутів можна представити витрати на вантажно-розвантажні роботи та складські витрати на збереження, як деякі проміжні шляхи і в такому випадку отримаємо можливість скористатися класичними алгоритмами пошуку оптимальних маршрутів.

Не вдаючись до кожного окремого випадку застосування того чи іншого алгоритму, загальні зміни, що можуть пристосовувати задачу виробничої логісти під класичні алгоритми пошуку мінімального шляху, можна представити наступним описом:

- кожна ланка, що пов'язує пункти наявного шляху, має бути представлена у вигляді відповідної послідовності з трьох, якими описуються (відповідно до порядку слідування): витрати на вантажні роботи з урахуванням об'єму постачання, витрати на переміщення об'єму постачання, витрати на розвантажувальні роботи з урахуванням об'єму постачання;
- опис кожного пункту повинен бути представлений, як ланка, що описує витрати на переміщення об'єму постачання між пунктами наявного шляху, проте нею представлені витрати на зберігання з урахуванням терміну зберігання на одиницю об'єму.

Загалом можемо представити задачу виробничої логістики, як задачу пошуку оптимального шляху за критерієм мінімізації витрат на доставку до одного виходу. Опис витрат при цьому представлений множинами витрат: на переміщення між пунктами; на вантажні роботи; на розвантажувальні роботи; на зберігання. Опис пунктів представлений множиною пунктів. Також наявні визначення: вихід – пункт доставки та термін доставки – час доставки.

Оскільки огляд проводиться з метою визначення можливості використання класичних методів та алгоритмів для розв'язання задачі виробничої логістики для використання при створенні програмного забезпечення, то виявлена адаптація дає змогу стверджувати, що така змога наявна.

В підсумок зазначимо, що внаслідок тезисного представлення проведених досліджень, опису бракує більшої деталізації, проте основна ідея викладена. За даних припущень, як результат, для адаптації пропонується використання алгоритмів Дейкстри та/або Беллмана-Форда (основа вибору - умови їх використання).

УДК 004

Р.О. Шевченко, *студент,* О.А. Кислун, *доцент,* Ю.М. Пархоменко, *доцент*
shevaroma31@gmail.com, kyslun@gmail.com, parhomenkoym@ukr.net
Центральноукраїнський національний технічний університет, Кропивницький

ОГЛЯД ПРОЦЕСУ ПІДТРИМКИ ПРАЦЕЗДАТНОСТІ ПК ПРИ ЙОГО ВИКОРИСТАННІ

Підтримка працездатності різноманітного обладнання при використанні є базовим процесом при організації виробництва незалежно від спрямування. Технічне обслуговування та ремонт завжди супроводжує використання складних пристроїв та систем. Основною проблемою при цьому є своєчасне виявлення некоректної роботи складових частин [1]. У відповідності до експлуатації ПК, за правильність відпрацювань останнього відповідає такий процес як моніторинг програмного та апаратного забезпечення (подальшому по тексту просто моніторинг). В даному опису проведеного дослідження ремонт ПК, як процес підтримки працездатності, лише зазначається, основну ж увагу приділено моніторингу.

Для розгляду моніторингу, як контролю працездатності ПК, обмежимся лише моніторингом за допомогою програмних засобів (апаратні засоби контролю поза межами даного розгляду), і при його загальноприйнятому поділу на системний моніторинг та моніторинг програмного забезпечення, зупинимося на системному моніторингу.

Загалом, системний моніторинг проводиться для контролю продуктивності роботи ПК та мереж, програмними засобами, яких на даний час розроблено не мало, це й вбудовані до операційної системи засоби адміністрування а також окремі сервісні програми та утиліти з різноманітним функціоналом діагностики (тестуванням) ПК в цілому та його окремих складових зокрема.

Сучасні ОС обладнані інструментами моніторингу системи та диспетчерами обладнання, а також засобами усунення проблем з працездатністю. Наприклад: ОС Windows обладнано засобом «Перевірка справності ПК» для підвищення продуктивності та усунення проблем із працездатністю устаткуванням [2].

В залежності від особистих вподобань, для відслідковування продуктивності та контролю працездатності фахівці можуть скористатися й засобами моніторингу та тестування, що не постачаються разом з ОС [3, 4]. Для аналізу наявного функціоналу, з погляду доцільності та спираючись на власні уподобання, досить значну множину доступних програмних рішень моніторингу ПК було скорочено до осяжного ряду з найбільш поширених, основні з них: AIDA64 (розробка FINALWIRE LTD) - застосунок для моніторингу та діагностики під ОС: Windows , Android , iOS , ChromeOS , Sailfish OS , Windows Phone , Ubuntu Touch та Tizen; SANDRA (розробка SISOFTWARE) - набір утиліт для моніторингу та діагностики під ОС: Windows, Android; CPU-Z (розробка CPUID) - утиліта збору інформація про пристрій з комплектуванням деякого тестуючого функціоналу під ОС: Windows, Android; OCCT (розробка OCBASE) - програмне забезпечення для тестування стабільності під ОС: Windows, Linux; WEBROOT SYSTEM ANALYZER (розробка WEBROOT SOFTWARE) - програмне забезпечення для тестування стабільності під ОС: Windows ; 1SIV (розробка RAY HINCHLIFFE) - утиліта збору інформація про локальну мережу та апаратне забезпечення ПК під ОС: Windows.

Розглянутий інструментарій направлений на вимірювання продуктивності та тестування працездатності, в окремих випадках наявні можливості налагодження - встановлення певних параметрів працездатності компонентів. Зауважимо, що для кращого сприйняття загалом наявні альтернативні способи представлення інформації. Не зважаючи на те, що наявні засоби досить в повній мірі охоплюють весь спектр робіт контролю роботи ПК, та все ж велика кількість вузькоспеціалізованих рішень та постійно зростаючі вимоги вказують на потребу все нових розробок. Для ефективного проектування такого програмного забезпечення під ОС Windows доцільним є використання бібліотеки WinSock API (WSA).

Список літератури

1. Організація робіт підприємств технічного обслуговування. Музичук В. І., Анісімов В. Ф. - Вінниця: ФОП Горбачук І.П., 2012. - 240с.

2.https://support.microsoft.com/uk-ua/windows//використання-програми-перевірка-справності-пк-9c8abd9b-03ba-4e67-81ef-36f37caa7844

3. https://www.imena.ua/blog/best-programs-for-computer-diagnostics/

4. https://daad.org.ua/3000-yak-pereviriti-produktivnist-kompyutera.html

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

VIII Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні технології"

24-25 квітня 2025 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