

УДК 004.45

DOI <https://doi.org/10.32689/maup.it.2024.2.10>

Роман ОНИЩЕНКО

студент кафедри інженерії програмного забезпечення та кібербезпеки,
Державний торговельний економічний університет, r.onyshchenko_fit_2m_23_m_z@knu.edu.ua
ORCID: 0009-0000-0492-900X

Наталія КОТЕНКО

кандидат педагогічних наук, доцент,
доцент кафедри інженерії програмного забезпечення та кібербезпеки,
Державний торговельний економічний університет, kotenkono@knu.edu.ua
ORCID: 0000-0002-2675-6514

Тетяна ЖИРОВА

кандидат педагогічних наук, доцент,
доцент кафедри інженерії програмного забезпечення та кібербезпеки,
Державний торговельний економічний університет, zhyrova@knu.edu.ua
ORCID: 0000-0001-8321-6939

**РОЛЬ ТА ЕФЕКТИВНІСТЬ ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ
В ТЕСТУВАННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Анотація. У статті досліджується роль та ефективність засобів штучного інтелекту в сучасних процесах тестування програмного забезпечення, основні напрямки його використання, а також його важливість для забезпечення надійності, безпеки та ефективності програмного забезпечення. З інтенсивним розвитком інтернет-технологій ефективна розробка веб-додатків є дуже важливою, що потребує впровадження нових методів у процес тестування програмного забезпечення.

Мета роботи: дослідити та узагальнити роль та ефективність інструментів штучного інтелекту у тестуванні веб-додатків в процесах сучасної розробки програмного забезпечення, визначити їх роль при забезпеченні надійності, безпеки та ефективності веб-застосунків.

Методологія: у дослідженні застосовується огляд інтернет-публікацій, літератури та аналіз існуючих засобів штучного інтелекту, які можна застосовувати у тестуванні програмного забезпечення.

Наукова новизна. Дослідження підкреслює необхідність застосування штучного інтелекту у тестуванні програмного забезпечення в умовах зростання вимог до ефективності тестування програмного забезпечення. Наголошено на важливості навчання тестувальників для застосування новітніх інструментів у роботі.

Висновки. Вимоги до ефективності сучасної розробки програмного забезпечення стають надзвичайно високими, що робить використання тільки мануального тестування з використанням класичних підходів малоефективним. Використання інструментів автоматизації тестування, застосування інструментів машинного навчання та штучного інтелекту є необхідним для підвищення ефективності процесів тестування та оптимізації ресурсів. Команди інженерів з розробки та тестування програмного забезпечення потребують застосування новітніх інструментів. Хмарні технології, інструменти DevOps та штучного інтелекту змінюють підходи до тестування. Завдяки росту доступності інструментів штучного інтелекту інженери можуть ефективніше працювати та забезпечувати високу ефективність тестування.

Ключові слова: штучний інтелект, машинне навчання, тестування програмного забезпечення, автоматизація тестування.

Roman ONYSHCHENKO, Nataliia KOTENKO, Tetyana ZHYROVA. THE ROLE AND EFFECTIVENESS OF ARTIFICIAL INTELLIGENCE TOOLS IN SOFTWARE TESTING

Abstract. The article investigates the role and effectiveness of artificial intelligence tools in modern software testing processes, the main areas of their application, and their significance for ensuring the reliability, security, and efficiency of software. With the rapid development of internet technologies, the effective development of web applications has become extremely important, necessitating the implementation of new methods in the software testing process.

Objective. To investigate and summarize the role and effectiveness of artificial intelligence (AI) tools in web application testing within modern software development processes, and to determine their role in ensuring the reliability, security, and efficiency of web applications.

Methodology. The study employs a review of internet publications and literature, as well as an analysis of existing artificial intelligence tools that can be applied in software testing.

Scientific Novelty. The study underscores the necessity of employing artificial intelligence in software testing in response to the increasing demands for efficiency in software testing. It highlights the importance of training testers to utilize the latest tools in their work.

Conclusions. The demands for efficiency in modern software development have become exceptionally high, making the use of only manual testing with classical approaches increasingly ineffective. The employment of test automation tools, along with the application of machine learning and artificial intelligence tools, is essential for enhancing the efficiency of testing processes

and optimizing resources. Development and testing engineering teams require the implementation of the latest tools. Cloud technologies, DevOps tools, and artificial intelligence are transforming testing approaches. With the growing availability of AI tools, engineers can work more efficiently and ensure high testing effectiveness.

Key words: artificial intelligence, machine learning, software testing, testing automation.

Вступ. Актуальність. В сучасному світі спостерігається подальше зростання залежності його процесів від інформаційних систем та технологій, тому розробка програмного забезпечення та вдосконалення технологій його тестування й досі є надзвичайно актуальними, як в Україні, так і у світі. Державним інституціям та бізнесу треба застосовувати сучасні, конкурентні інструменти розробки та тестування програмного забезпечення у своїй діяльності. Останніми роками спостерігається стрімкий розвиток та застосування технологій штучного інтелекту (далі – ШІ) в усіх сферах діяльності людини. Актуальність статті полягає у необхідності обґрунтування вдосконалення процесів тестування програмного забезпечення із застосуванням технологій ШІ.

Мета статті полягає в дослідженні перспектив використання технологій ШІ у тестуванні програмного забезпечення. Визначення шляхів вдосконалення навичок фахівців з тестування у частині застосування технологій ШІ.

Аналіз досліджень і публікацій. Розгляду технологій тестування програмного забезпечення присвячені праці таких авторів як Niranjana Murthy M., Khaiyum S., Rakshitha K. P. [14], Jorgensen P., Vries B. D. [13], Badgett T., Myers G. J., Sandler C. [10], Copeland L. [12], Beizer B. [11] та ін. Ці праці стали класичними для багатьох поколінь тестувальників.

Останнім часом почали з'являтися праці, які висвітлюють питання ШІ та автоматизації тестування. Наприклад, у книзі "Artificial intelligence and software testing" [9] йдеться про основні поняття і методи ШІ, що можуть бути використані у тестуванні програмного забезпечення. Це включає машинне навчання, обробку природної мови та інші алгоритми ШІ. Проте слід зазначити, що тема обрано дослідження потребує додаткового висвітлення.

Виклад основного матеріалу. Під ШІ слід розуміти систему, створену за допомогою інформаційних технологій, що намагається моделювати низку аспектів ментальних процесів та функціонування людини. Термін «ШІ» часто використовується для опису систем, які здатні емулявати основні функції сприйняття та розуміння, що є характерними для людської ментальності [1].

Машинне навчання – це галузь ШІ, яка використовує алгоритми та дані для імітації способу навчання людей, поступово покращуючи точність.

Нейронні мережі є фундаментальною технологією для ШІ. Нейронні мережі – це спроба змоделювати роботи мозку живого організму, який складається з мільйонів нейронів, кожен з яких з'єднаний з кількома іншими. Кожен окремий нейрон дуже простий, але разом вони здатні навчатися виконувати складні завдання. Штучні нейронні мережі імітують ці процеси, покладаючись на навчальні дані, з часом підвищуючи свою точність.

Глибинне навчання – сфера штучного інтелекту, яка дозволяє машинам робити ті задачі, які зазвичай роблять люди. Така техніка машинного навчання дозволяє комп'ютерам вчитися на людських прикладах, що в результаті допомагає автоматизувати різноманітні процеси.

Тестування займає важливе місце у процесі створення програмного забезпечення. Зазвичай виділяють такі етапи тестування програмного забезпечення [4]:

1. Планування та управління, аналіз та проектування;
2. Впровадження та реалізація;
3. Оцінка критеріїв виходу і написання звітів;
4. Дії по завершенню тестування.

Розглянувши низку інтернет-публікацій щодо впровадження технологій ШІ у процес тестування програмного забезпечення, можна відмітити наступне.

ШІ дозволяє автоматизувати тестування. Дозволяє знизити кількість одноманітних завдань, а тестувальникам зосередитись на вагоміших аспектах тестування. ШІ також покращує виконання тестувальних функцій, в порівнянні зі звичайними інструментами автоматизації. Зазначається, що алгоритми ШІ можна навчити знаходити, розпізнавати та аналізувати великі масиви даних, на що не здатна людина, а це пришвидшує процес тестування, знижує витрати, та підвищує ефективність [3].

ШІ покращує якість тестування. З його використанням можна автоматизувати тестування та забезпечити глибший та всебічніший аналіз продукту [5].

У роботі мануального тестувальника ШІ може допомогти із:

1. Аналізом вимог, генерацією чек-листів.
2. Написанням тест-кейсів для тестування.

3. Генерацією тестових даних.
4. Створенням тестової документації (тест-планів, тест-стратегій, тестових звітів).
5. Створенням діаграм.

Однак, дуже важливою є проблема промптів (prompt). Для того, щоб інструмент ШІ згенерував корисну відповідь, необхідно правильно зробити запит [2].

Зазначається, що сучасні devops-практики (CI/CD) розвинули ландшафт тестування програмного забезпечення, створивши багато можливостей для застосування технологій ШІ. Застосування ШІ в тестуванні програмного забезпечення в основному зосереджується на вирішенні двох поширених сценаріїв: недостатня або надмірна кількість тестів. Відсутність або недостатність тестів створює ризик, оскільки розробники можуть не виявити помилки. Іноді трапляється протилежна ситуація. Організації часто стикаються з роздутими тестовими потоками через надмірну кількість тестів. Це вузьке місце значно перешкоджає ефективності та продуктивності тестування. ШІ приходить на допомогу, забезпечуючи широкомасштабне тестування за допомогою моделей машинного навчання (ML), які визначають критичні тести, оптимізують вибір. У той час як різні підходи аналізують зміни коду, щоб оцінити їх вплив на тести, прогнозний вибір тестів представляє передовий підхід ШІ порівняно з ручним аналогом аналізу впливу тестів. Використовуючи моделі ML, навчені на основі результатів історичних тестів, ШІ визначає необхідні тести для виконання певної зміни коду. Це дозволяє командам тестувальників досягати швидших циклів тестування та підвищеної надійності коду, дозволяючи розробникам зосередитися на інноваціях, прискорюючи цикли випуску [15].

Використання ШІ в автоматизованому тестуванні значно розширює можливості тестувальників, автоматизуючи не тільки аналітичні, але й повсякденні рутинні завдання. Інтелектуальні системи можуть бути налаштовані на автоматичну генерацію тестових даних, що значно знижує час, необхідний для підготовки тестування, та водночас збільшує його охоплення і глибину. Ці системи можуть враховувати численні варіанти вхідних даних та їх комбінації, які було б неможливо обробити вручну, тим самим підвищуючи якість тестування. ШІ може автоматизувати процес виявлення та класифікації помилок, визначаючи їхню критичність та пріоритетність, що дозволяє розробникам швидше зосередитися на найбільш значущих дефектах. Автоматизовані системи можуть відслідковувати зміни у кодї і самостійно адаптувати тестові набори до цих змін, гарантуючи актуальність тестів та високу релевантність результатів [6].

Вважається, що ШІ може аналізувати вимоги до програмного забезпечення і автоматично генерувати тести для перевірки функціональності застосунку або компонента. Це значно прискорює процес їхньої підготовки і забезпечує повніше покриття. ШІ також може використовувати аналітику і машинне навчання для побудови ефективних стратегій тестування, які враховують різні фактори: історію помилок, специфікацію вимог, архітектуру системи тощо.

Крім того, ШІ здатен автоматично створювати сценарії тестів на основі вимог користувачів, забезпечуючи відповідність продукту їхнім очікуванням і працюючи коректно в реальних умовах. Також ШІ може генерувати автоматизовані тестові скрипти для перевірки функціональності та нефункціонального тестування [7].

Отже, можна виділити такі переваги ШІ в тестуванні програмного забезпечення:

- Збільшення покриття тестами.
- Пришвидшення виконання тесту.
- Задоволення потреб безперервного тестування.
- Вдосконалення пріоритетності тестів.
- Підвищення ступеню точності тестів.
- Зменшення тестового обслуговування.
- Зменшення вартості тестування.

Автор дослідження [8] вказує також на наступні приклади використання ШІ в тестуванні (рис. 1).

Дослідивши низку публікацій слід зазначити наступне. Найбільш перспективними напрямками застосування штучного інтелекту в тестуванні програмного забезпечення будуть такі напрямки: допомога у рутинній роботі тестувальника (підготовка тест-кейсів, аналіз вимог, генерація тестових даних), аналіз великих обсягів інформації, підготовка оптимальної кількості тестів, генерація тестових скриптів для перевірки функціональності та нефункціонального тестування, визначення пріоритетності тестів.

Перевагами застосування ШІ є збільшення точності тестів, покращення точності процесу тестування, пришвидшення виконання, покращення пріоритезації тестів, безперервне тестування, економія витрат.

- Слід вказати на такі проблеми впровадження ШІ у процеси тестування програмного забезпечення:
1. Дефіцит персоналу, здатного ефективно втілювати інструменти штучного інтелекту у тестування програмного забезпечення.
 2. Використання застарілих систем, які можуть бути несумісними з інструментами тестування на основі ШІ.
 3. Тестування за допомогою ШІ значною мірою покладається на високоякісні дані для отримання точних результатів. Якщо дані є неадекватними або помилковими, це може призвести до неточних результатів тестування.
 4. Тестування за допомогою ШІ є складним, і робота системи ШІ може бути непрозорою для користувачів.
 5. Застосування ШІ в процесах тестування програмного забезпечення може бути дорогавартісним, як в плані начального інвестування, так і з огляду на постійні витрати.
 6. Системи для ШІ, які використовуються для тестування програмного забезпечення, можуть бути вразливими до атак з боку хакерів, що створює додаткові проблеми безпеки.

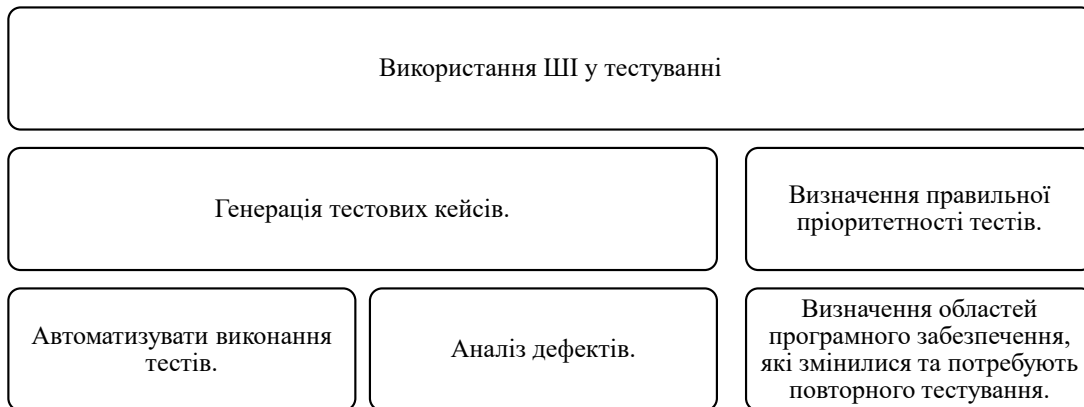


Рис. 1. Використання ШІ в тестуванні програмного забезпечення

Висновки. Застосування технологій ШІ в роботі тестувальника надає можливість оптимізації та покращення без значних вкладень у ресурси чи персонал. В свою чергу це надає можливість випускати програмні продукти більш високої якості швидше. А це покращує досвід клієнтів, та відповідно прибутки від застосування інформаційних технологій в бізнесі або державному секторі.

Проте слід зазначити, що як автоматизація тестування має бути економічно обґрунтованою, так і застосування методів ШІ в тестуванні мають бути економічно обґрунтованими, оскільки існує низка проблем щодо застосування ШІ інтелекту в тестуванні програмного забезпечення. Організаціям потрібен ефективний спосіб впровадження цих потужних технологій у практику тестування програмного забезпечення.

Для того, щоб відповідати потребам ринку праці тестувальникам вже зараз необхідно опанувати сучасні інструменти ШІ, а приватні та державні заклади освіти повинні впроваджувати у свої освітні програми зазначені інструменти.

Список використаних джерел:

1. Дослідження застосування штучного інтелекту у кібербезпеці / О. І. Голубенко та ін. ITSynergy. 2023. № 2. С. 71–81. URL: <https://doi.org/10.53920/its-2023-2-5> (дата звернення: 28.06.2024).
2. Тестувальники, які володіють інструментами ШІ, замінять тих, хто їх не використовує. Anywhere Club. URL: <https://aw.club/global/uk/blog/how-to-use-artificial-intelligence-in-testing> (дата звернення: 27.06.2024).
3. Тестування програмного забезпечення з використанням штучного інтелекту. Delivering excellence with professionals at Brainberry.ua. URL: <https://brainberry.ua/uk/newsroom/blog/software-testing-using-ai> (дата звернення: 27.06.2024).
4. Фундаментальний процес тестування - QALight. QALight. URL: <https://qalight.ua/baza-znaniy/fundamentalnij-protses-testuvannya/> (дата звернення: 27.06.2024).
5. Штучний інтелект та програмне забезпечення: плюси від інтеграції. Об'єднання Intecracy Group. URL: <https://intecracy.com/ua/news/shtuchnyi-intelekt-ta-programne-zabezpechennia-pliusy-vid-intehratsii.html/> (дата звернення: 27.06.2024).
6. Штучний інтелект у QA: майбутнє автоматизованого тестування. Largest HQ. URL: <https://largesthq.com/shtuchnyy-intelekt-u-qa-maybutne-avtomatyzovanoho-testuvannya/> (дата звернення: 27.06.2024).

7. Що може робити ШІ на вашому проєкті – горизонтально і вертикально. Досвід архітектора. Dou. URL: <https://dou.ua/forums/topic/44863/> (дата звернення: 27.06.2024).
8. Як AI змінить тестування програмного забезпечення - Visure Solutions. Visure Solutions. URL: <https://visuresolutions.com/uk/blog/ways-ai-will-change-software-testing/> (дата звернення: 27.06.2024).
9. Artificial intelligence and software testing: a practical guide to quality / J. Davenport et al. BCS Learning & Development Limited, 2022.
10. Badgett T., Myers G. J., Sandler C. Art of software testing. Wiley & Sons, Incorporated, John, 2011. 256 с.
11. Beizer B. Black-Box testing: techniques for functional testing of software and systems. Wiley & Sons, Incorporated, John, 2008. 320 с.
12. Copeland L. A practitioner's guide to software test design. Boston, Mass: Artech House, 2004. 294 с.
13. Jorgensen P., Vries B. D. Software testing. Taylor & Francis Group, 2021.
14. Niranjanamurthy M., Khaiyum S., Rakshitha K. P. Trends in software testing. Wiley & Sons, Incorporated, John, 2022.
15. NIXSolutions: як штучний інтелект революціонує QA. Nix Solutions | Nix Solutions. URL: <https://nixsolutions-qa.com/ai-revolutionizing-qa/> (дата звернення: 27.06.2024).