

ARTIFICIAL INTELLIGENCE FOR AUTOMATED TESTING AND BUG TRACKING

Halan O.*Candidate of Technical Sciences, Associate Professor of IT, State University of Infrastructure and Technology (Kyiv, Ukraine).***Buhai D.***Second-level (educational and scientific) higher education student of the State University of Infrastructure and Technology (Kyiv, Ukraine)***ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ТА ВІДСТЕЖЕННЯ ПОМИЛОК****Галан О.В.***Кандидат технічних наук, доцент каф ІТ, Державний університет інфраструктури та технологій (Київ, Україна)***Бугай Д.О.***Здобувач вищої освіти другого (освітньо-наукового) рівня Державного університету інфраструктури та технологій (Київ, Україна)*<https://doi.org/10.5281/zenodo.13863436>**Abstract**

The article is dedicated to the use of artificial intelligence for software testing automation and bug tracking. It describes the key aspects of implementing intelligent testing systems, including automatic test scenario generation, code analysis, defect prediction, and log analysis. Specific examples of tools that demonstrate an increase in the speed and efficiency of testing are considered. The article provides statistical data confirming the growing popularity of the use of intelligent systems in software testing among large companies, which allows improving product quality and optimising processes.

Анотація

Стаття присвячена використанню штучного інтелекту для автоматизації тестування програмного забезпечення та відстеження помилок. Описуються ключові аспекти впровадження інтелектуальних систем тестування, що включають автоматичну генерацію тестових сценаріїв, аналіз коду, прогнозування дефектів та аналіз логів. Розглянуто конкретні приклади інструментів, які демонструють підвищення швидкості та ефективності тестування. Наведені статистичні дані, які підтверджують зростання популярності використання інтелектуальних систем у тестуванні програмного забезпечення серед великих компаній, що дозволяє покращити якість продуктів і оптимізувати процеси.

Keywords: artificial intelligence, test automation, software testing, bug tracking, machine learning, intelligent systems, code analysis, test scenario generation, defect prediction, software product quality assurance.

Ключові слова: штучний інтелект, автоматизація тестування, тестування програмного забезпечення, відстеження помилок, машинне навчання, інтелектуальні системи, аналіз коду, генерація тестових сценаріїв, прогнозування дефектів, забезпечення якості програмного продукту.

Актуальність проблеми. Процес розробки програмного забезпечення в сучасному світі є ресурсоємним і складним, поєднуючи в собі всі етапи життєвого циклу продукту: від планування та проектування до самого процесу розповсюдження та підтримки [1]. Одне з важливих місць займає тестування, яке покликане гарантувати високу якість продукту шляхом мінімізації кількості помилок. Серйозним недоліком такого підходу є те, що ручне тестування може затягнутися на дуже довгий час, що вносить свій внесок у виникнення помилок обумовлених людським фактором [2]. Автоматизація тестування програмного забезпечення останнім часом перетворилася на важливий інструмент, який дозволяє скоротити час, підвищити точність і збільшити ефективність [3]. Завдяки розвитку штучного інтелекту з'явилася можливість застосовувати інтелектуальні системи автоматизації тестування, завдяки чому виявлення дефектів значно покращується і передбачає менше втручання людини.

Що таке інтелектуальна система автоматизації тестування? До інтелектуальних систем автоматизації тестування відносяться рішення, які використовують штучний інтелект і машинне навчання в процесах пошуку і відстеження помилок, автоматизації процесів тестування [4]. Такі системи можуть адаптуватися до змін у програмному забезпеченні, виявляти приховані помилки та надавати глибокий аналіз продуктивності системи. Це дозволяє тестувальникам і розробникам зосередитися на більш складних завданнях, зменшивши кількість рутинних операцій.

Використання штучного інтелекту для автоматизованого тестування. Штучний інтелект і машинне навчання дозволяють автоматизувати в області тестування наступні процеси [5].

1. Генерацію тестових випадків. За допомогою штучного інтелекту можна проаналізувати вимоги і згенерувати тестові сценарії, що включають кілька

варіантів дій користувачів [6]. Таким чином, підготовка тестів займає набагато менше часу, а покриття тестів збільшується.

2. Тестування користувацького інтерфейсу. Інструменти зі штучним інтелектом виконують автоматизоване тестування користувацького інтерфейсу, імітуючи взаємодію користувача з програмним продуктом [7]. Ці інструменти призначені для пошуку проблем у функціонуванні заданих елементів інтерфейсу.

3. Прогнозування дефектів. За допомогою машинного навчання, аналізуючи попередні дані про помилки, можна передбачити, які частини коду мають високу ймовірність розміщення дефектів [8]. Це дозволить розробникам сконцентруватися на критично важливих частинах системи.

4. Аналіз коду та тестів. Інтелектуальні системи можуть автоматично аналізувати код для виявлення потенційних вразливостей, відхилень від стандартів і структурних проблем. Штучний інтелект також здатний аналізувати результати тестів, щоб набагато швидше визначити, що йде не так і де це відбувається.

Відстеження помилок за допомогою штучного інтелекту. Відстеження помилок - одна з основних частин процесу забезпечення якості програмного продукту. Штучний інтелект може значно збагатити цей процес завдяки кільком функціям. Наприклад, системи штучного інтелекту можуть автоматично аналізувати системні журнали і виявляти потенційні вразливості або помилки, які не були виявлені за допомогою людського аналізу [9]. Це значно прискорює процес виявлення критичних проблем. Ще однією важливою перевагою штучного інтелекту є можливість класифікувати та визначати пріоритетність помилок на основі їхньої критичності, що дозволяє командам тестувальників та розробників зосередитися на вирішенні найбільш важливих завдань. Системи штучного інтелекту також значною мірою сприяють автоматизації процесу аналізу першопричин, що значно економить час при спробі виправити помилку. Це дуже корисно у випадках складних помилок, коли ручний пошук першопричини може зайняти багато часу. Його переваги полягають у тому, що інтелектуальна система вчиться на попередніх виправленнях і застосовує знання в нових випадках. Це прискорює процес виправлення помилок і допомагає постійно вдосконалювати систему з кожним новим етапом відстеження та усунення проблем.

Переваги використання штучного інтелекту в тестуванні та відстеженні помилок. Однією з переваг використання штучного інтелекту у відстеженні помилок є те, що він дає значний приріст швидкості під час тестування. Тестування, що виконується за допомогою інтелектуальних систем, дозволяє підвищити продуктивність порівняно з ручним тестуванням, оскільки сотні і тисячі тестів можуть бути автоматично виконані в режимі реального часу. Крім того, штучний інтелект підвищує точність під час тестування, а отже, зменшує вплив людського фактору і робить процес перевірки програмного продукту інтенсивним і всебічним. Ще

одна важлива перевага полягає в тому, що інтелектуальні системи автоматично отримують інформацію про будь-які зміни в програмному забезпеченні, отже, їхні стратегії тестування адаптуються в реальному часі. Це позбавляє команду від рутинної роботи, дозволяє їм зосередитися на творчих і критичних аспектах розробки та значно підвищує продуктивність праці [10].

Методи, що застосовуються для реалізації інтелектуальних систем. Інтелектуальні системи автоматизації тестування програмного забезпечення використовують різноманітні методи й підходи для забезпечення точності, швидкості та масштабованості процесу тестування. Ось кілька конкретних методів та прикладів їхнього використання в реальних сценаріях.

1. Одним з найважливіших способів застосування штучного інтелекту в автоматизованому тестуванні є машинне навчання для аналізу коду і прогнозування дефектів. Приклади включають використання глибокого навчання в системах на кшталт Diffblue Cover [11] для генерації тестів на основі коду та історичних даних про дефекти. Згідно з експериментами, таким системам вдалося зменшити кількість критичних помилок на 30-40%, а також прискорити тестування на 20-25% порівняно з ручним підходом.

2. Генерація тестових сценаріїв за допомогою штучного інтелекту. Інструменти на кшталт digital.ai [12] автоматично генерують тестові кейси мобільних додатків шляхом сканування інтерфейсів і поведінки користувачів. Система може генерувати тести для більш ніж 80% тестового покриття інтерфейсу користувача за лічені хвилини, скорочуючи час підготовки сценаріїв в середньому на 2-3 дні, якщо це робиться вручну.

3. Автоматизація регресійного тестування. Регресійне тестування – це повторне виконання тестів, яке дозволяє бути впевненим, що зміни в коді не призвели до появи нових помилок. Автоматизувати регресійне тестування можна за допомогою розумних інструментів, одним з яких є Functionize [13]. Система здатна самостійно адаптуватися до змін у коді, автоматично оновлюючи тестові випадки. Завдяки використанню Functionize продуктивність тестування зросла на 60%, а кількість виправлень вручну зменшилася на 50%.

4. Аналіз поведінки користувачів і прогнозування вразливостей. Завдяки штучному інтелекту можна аналізувати активність реальних користувачів і передбачати точки збою, які можуть виникнути. Наприклад, Appliflow [14] виконує автоматичний аналіз змін на веб-сторінках для виявлення прихованих проблем - наприклад, графічних помилок, які людське око навряд чи зможе помітити. За допомогою цієї системи вдалося зменшити кількість пропущених візуальних дефектів на 45% [15].

Аналіз логів та аналіз першопричин. Інтелектуальні платформи, такі як Splunk і Logz.io, безперервно відстежують логи додатків, заздалегідь передбачаючи події, які можуть призвести до аварій. Класифікуючи помилки за допомогою штучного інтелекту, ці інструменти дають вам можливість визначити першопричини з системного журналу.

Наприклад, ця система скорочує середній час виявлення та усунення проблем на 35%.

6. Тестування на основі моделей. Серед підходів, які може використовувати штучний інтелект, є тестування на основі моделей. Система аналізує бізнес-логіку програми і створює тести на основі моделей поведінки користувачів або на основі очікуваних результатів роботи системи. За допомогою цього методу можна зменшити кількість ручної роботи на етапі розробки тестових кейсів на 50-70%, що може бути особливо корисним у великих системах з розгалуженою логікою, таких як ERP або CRM-системи.

Статистичні дані. У звіті, наданому Capgemini у 2023 році, було зазначено, що 48% великих організацій вже використовують штучний інтелект в процесах тестування програмного забезпечення, а ще 32% зроблять це протягом двох років. У компаніях, які використовують інтелектуальні системи для автоматизації тестування, швидкість випуску оновлень зросла з 25% до 40%, а кількість помилок у виробничому середовищі скоротилася на 30-50% [16].

Використання штучного інтелекту дозволяє суттєво підвищити ефективність автоматизації тестування, зробивши цей процес ще більш гнучким, швидким і надійним. Це допоможе не тільки в підвищенні якості продукту, але і в оптимізації ресурсів, що має серйозне значення в умовах стрімкого розвитку технологій і підвищення вимог до якості програмного забезпечення.

Приклад успішного впровадження. Компанія Google активно застосовує штучний інтелект для автоматизації тестування своїх продуктів. Згідно з дослідженням компанії, за допомогою інтелектуальних систем крос-платформне тестування браузера Chrome було прискорено на 20%, а витрати на тестування скоротилися на 15%. При цьому точність виявлення критичних дефектів зросла на 30%, що значно зменшило кількість екстрених випусків оновлень [17].

Виклики впровадження штучного інтелекту в автоматизацію тестування. Незважаючи на значні переваги використання штучного інтелекту в автоматизації тестування програмного забезпечення, існує також кілька проблем, які можуть супроводжувати його впровадження. Одним із головних є висока вартість впровадження, оскільки інтелектуальні системи можуть бути досить дорогими, особливо на початковому етапі. У випадку з невеликими компаніями це є стримувальним фактором від впровадження таких систем. Крім того, робота зі штучним інтелектом вимагає технічних навичок - дуже висококваліфікованих фахівців з машинного навчання та тестування програмного забезпечення. Ще однією проблемою є якість і кількість тестових даних, оскільки від цього залежить ефективність роботи інтелектуальних систем, а їх кількість може бути обмеженою [18].

Висновки. Інтелектуальна автоматизація тестування програмного забезпечення - це величезний стрибок вперед для забезпечення якості програмного забезпечення. Вона може значно підвищити ефективність тестування, зменшити кількість

помилки і заощадити час на їх пошук. Хоча впровадження штучного інтелекту вимагає значних ресурсів і технічних знань, потенційні вигоди від його використання перевищують витрати, роблячи його важливою частиною сучасної розробки програмного забезпечення.

Список літератури:

1. Gerardus Blokdyk. Software Development Lifecycle a Complete Guide, 2019.
2. Castaneda, S. (2016). The Manual Testing Handbook - Everything You Need to Know about Manual Testing. Emereo Pty Limited.
3. Dustin E. Automated software testing: introduction, management, and performance/ Dustin E, Rashka J., Paul J. Addison Wesley, 1999.
4. Буров Є.В. Інтелектуальна система автоматизованого тестування програмного продукту з використанням алгоритмічних моделей / Національний університет "Львівська політехніка", 2011.
5. AI-Infused Test Automation: Revolutionizing Software Testing through Artificial Intelligence. OrangeBooks Publication, 2023.
6. Artificial intelligence and data science engineering: SK Research Group of Companies, 2024.
7. Li, P., Huynh, T., Reformat, M., Miller, J.: A practical approach to testing GUI systems.
8. Winteringham, M. Software Testing with Generative AI. Manning, 2024.
9. How to Develop a QA Strategy with Generative AI: Bosun Sokeke, 2024.
10. Kinsbruner, E. (2020). Accelerating Software Quality: Machine Learning and Artificial Intelligence in the Age of DevOps: Amazon Digital Services LLC - KDP Print US.
11. Jay, Rabi. Enterprise AI in the Cloud: A Practical Guide to Deploying End-to-End Machine Learning and ChatGPT Solutions. Сполучені Штати Америки, Wiley, 2023.
12. Digital.AI URL: <https://digital.ai/>
13. Functionize: AI Test Automation with Machine Learning. URL: <https://www.functionize.com/>
14. Applitools: AI-Powered End-to-End Testing Platform. URL: <https://applitools.com>.
15. Artificial Intelligence in HCI: 5th International Conference, AI-HCI 2024, Held As Part of the 26th HCI International Conference, HCII 2024, Washington, DC, USA, June 29 - July 4, 2024, Proceedings, Part I.. Швейцарія, Springer International Publishing AG, 2024.
16. Тягунова М. Ю., Киричек Г. Г., Костецький Д. В. Розрахунок економії часу при створенні тесту за допомогою штучного інтелекту. Systems and technologies. 2024. Т. 67, № 1. С. 65–71. URL: <https://doi.org/10.32782/2521-6643-2024-1-67.10>
17. Why we focus on AI. URL: <https://ai.google/why-ai/>
18. Kateryna Ivanova, Galyna Kondratenko, Ievgen Sidenko, Yuriy Kondratenko: Artificial Intelligence in Automated System for WebInterfaces Visual Testing. URL: <http://ceur-ws.org/Vol-2604/paper68.pdf>