

ЗАСТОСУВАННЯ ШІ В АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Ларченко С. О.

Науковий керівник – к.т.н., доц. Лановий О. Ф.

Харківський національний університет радіоелектроніки, каф. ПІ

м.Харків, Україна

e-mail: serhii.larchenko.cpe@nure.ua

This work focuses on the latest techniques for applying artificial intelligence (AI) in software test automation, in particular, the ability to generate test scenarios using AI. The focus is on using machine learning technologies to improve the automation and efficiency of testing processes. Challenges and prospects for further research in this area are discussed, with an emphasis on the search for the latest technologies for continuous improvement of the quality and efficiency of automated software testing.

Автоматизація процесу тестування стала невід'ємною частиною сучасних практик в індустрії розробки програмного забезпечення. Існує значна кількість методів та технологій контролю за якістю та надійністю ПЗ [1]. Автоматизовані тести можуть бути виконані швидко та ефективно та здатні виконувати багато операцій одночасно і працювати безперервно. Виконання автоматизованого тестування визначає, чи було отримано відповідні результати, та чи може бути проведено об'єднання призвести до успіху або збою інтеграції [2]. Машинне навчання (ML) та інші технології штучного інтелекту (ШІ) можуть сприяти ширшому покриттю тестування і виявленню багів, які можуть знизити рівень мануального тестування. Нарешті, автоматизоване тестування має важливе значення для збереження ресурсів.

В останні роки штучний інтелект з генерацією (GenAI) привернув увагу та зацікавленість у сфері програмної інженерії. Інструменти GenAI, такі як GitHub Copilot і ChatGPT, були швидко досліджені в різних професійних контекстах.

Дослідження використання ШІ для автоматизованого тестування спрямоване на визначення оптимальних стратегій застосування штучного інтелекту для створення автоматизованих тестових сценаріїв, для подальшого використання в розробці тестового фреймворку.

Було розглянуто п'ять методів ШІ таких як: Machine Learning, Neural Networks, Support Vector Machines, Natural Language Processing, Logic Programming. Для визначення кращого методу було використано введення багатокритеріальної задачі вибору, яка передбачає наявність кількох конкуруючих критеріїв, які мають бути оптимізовані для досягнення найкращого результату у контексті цільової системи. Аналіз критеріїв для багатокритеріального вибору допомагає об'єктивно оцінити кожен із

варіантів, враховуючи різноманітні фактори, що впливають на вибір методу.

Методи ML, було обрано кращим для подальшого дослідження, тому що він дозволяє системам вчитися і зростати з досвіду, поступово покращуючи свою продуктивність і ефективність. Методи ML підвищують ефективність життєвого циклу програмної інженерії (SE) [3]. В автоматизації тестування, ML часто використовується для предиктивного аналізу, ідентифікації моделей та змінних, що викликають помилки, та автоматичного генерування сценаріїв тестування.

На етапі експерименту, будуть розглянуті можливості генерації автоматичних тестів, за допомогою Large Language Model (LLM), використовуючи User story, як основу для генерації авто-тесту. Для генерації авто-тестів будуть використовуватись такі мовні моделі як GPT-3.5, GPT-4, Bart.

Експерименти допоможуть зібрати інформацію про якість створених тестів, швидкість, точність виконання запиту, також на сформованих кожною моделлю наборах тестів, буде проведено порівняння в швидкості виконання сформованих авто-тестів, час знаходження помилки, оцінка придатності тесту для пошуку дефекту.

Основні висновки використання ШІ в автоматизації тестування, полягають у виявленні можливостей для підвищення якості програмного забезпечення, шляхом генерації тестових сценаріїв, зменшення часу, необхідного для проведення тестів, та збільшення точності виявлення дефектів. Крім того, це може сприяти покращенню розробки та впровадженню нових стратегій тестування, що в свою чергу призведе до постійного розвитку галузі автоматизованого тестування.

Список використаних джерел:

1. Лановий О. Ф. Візуалізація в методах тестування програмного забезпечення: thesis. 2017. URL: <http://openarchive.nure.ua/handle/document/9432> (дата звернення: 02.03.2024).

2. Фундукян А. Метод розстановки пріоритетів тестів. education and science of today: intersectoral issues and development of sciences / chair О. Лановий. 2021. URL: <https://doi.org/10.36074/logos-19.03.2021.v2.31> (дата звернення: 29.02.2024).

3. Kotti Z., Galanopoulou R., Spinellis D. Machine Learning for Software Engineering: A Tertiary Study. ACM Computing Surveys. 2022. URL: <https://doi.org/10.1145/3572905> (date of access: 29.02.2024).