

УДК 004.85

**ДОСЛІДЖЕННЯ МОДЕЛЕЙ ВИКОРИСТАННЯ ЗАСОБІВ
МАШИННОГО НАВЧАННЯ ДЛЯ ГЕНЕРАЦІЇ
ТЕСТ-КЕЙСІВ У ІТ-ПРОЄКТАХ**

Альошкін О.А.

Науковий керівник – д.т.н., професор кафедри ІУС Євланов М.В.
Харківський національний університет радіоелектроніки, каф. ІУС
м. Харків, Україна

тел.: +38(095) 441-04-71, e-mail: oleksii.aloshkin@nure.ua

This work is focused on research of the application of artificial intelligence for generating test cases in IT projects. With the growing complexity of software development, automated solutions are needed to ensure code quality and efficient testing. One of the jobs where machine learning can help is generating test cases. Two methods for generating test cases, genetic algorithm and Q-learning, are examined and compared in this work. The results highlight their potential benefits and limitations of using these algorithms.

За останні роки використання штучного інтелекту поширилось у різних галузях. Однією зі сфер, де штучний інтелект виявився особливо цінним, є процеси забезпечення якості ІТ-проектів [1].

Оскільки розробка програмного забезпечення стає дедалі складнішою, зростає потреба в автоматизованих інструментах для забезпечення якості коду та ефективності тестування. Штучний інтелект може допомогти у виконанні цього процесу, за допомогою аналізу величезних обсягів даних та виявлення закономірностей, які тестувальники можуть не помітити. Створення тестових кейсів є важливою частиною тестування програмного забезпечення і передбачає створення вхідних даних та очікуваних результатів для певного компонента системи.

Метою цієї роботи є дослідження моделей використання засобів машинного навчання для розв'язання задачі генерації тест-кейсів. Розглянуто 2 методи для створення тест-кейсів: генетичний алгоритм та Q-навчання. Виконано порівняння цих алгоритмів у контексті створення тест-кейсів.

Генетичний алгоритм – це тип еволюційного алгоритму, який використовує евристичний підхід. Аналогічно тому, за яким у природі еволюціонують усі види живих організмів і проводиться генетичний відбір найбільш пристосованих до неї особин. Вхідні дані задачі представлені популяцією тестових кейсів, яка проходить через оператори генетичного алгоритму, тобто схрещування та мутації. Початкова популяція може бути обрана вручну або випадковим чином та являє собою рядок двійкових або дійсних чисел, кожна цифра в якому називається геном.

Цей процес триває до тих пір, поки не буде досягнутий бажаний рівень покриття тестування [2].

Q-навчання є різновидом безмодельної техніки навчання з підкріпленням. Його алгоритми навчаються за допомогою методу спроб та помилок при взаємодії з навколишнім середовищем. Він заснований на концепції поведінкової психології, що відіграє ключову роль у штучному інтелекті. Агент отримує винагороду, якщо досягає мети. Q-навчання може сходитися до оптимального рішення, але для цього потрібно багато ітерацій. Алгоритм може працювати в динамічних і складних середовищах, а також адаптуватися до змін [3].

Генетичні алгоритми ефективно знаходять різноманітні рішення і добре підходять для складних оптимізаційних задач. Однак, вони можуть бути обчислювально дорогими, особливо коли існує великий простір пошуку. На відміну від них, Q-навчання є більш ефективним з точки зору обчислень, але він не завжди може знайти оптимальне рішення, особливо в складних системах.

Таким чином, Q-навчання та генетичні алгоритми – це алгоритми, які використовують різні підходи для розв’язання задачі:

- Q-навчання розвивається через взаємодію з навколишнім середовищем методом спроб і помилок, в той час як генетичний алгоритм розвиває популяцію тестових кейсів з плином часу;
- Q-навчання вимагає чітко визначеного простору станів, простору дій та функції винагороди, в той час як генетичний алгоритм вимагає чітко визначеної функції пристосованості;
- Q-навчання може працювати в динамічних і складних середовищах, тоді як генетичний алгоритм може боротися з ними.

В результаті дослідження було визначено сильні та слабкі сторони використання генетичного алгоритму та Q-навчання для генерації тест-кейсів у IT-проєктах. Проведено порівняння алгоритмів між собою та визначено доцільність використання.

Список використаних джерел:

1. Trudova, A., Dolezel, M., & Buchalceva, A. (2020). Artificial Intelligence in Software Test Automation: A Systematic Literature Review. 15th International Conference on Evaluation of Novel Approaches to Software Engineering, м. Prague, Czech Republic. <https://doi.org/10.5220/0009417801810192>
2. Sharma, S., & Bhatia, S. (2018). Comparative Study of Software Testing Through Genetic Algorithm. International Journal of Computer Science and Information Security, 16, (10), 27–33.
3. Yasin, H. N., Hamid, S. H. A., & Raja Yusof, R. J. (2021). DroidbotX: Test Case Generation Tool for Android Applications Using Q-Learning. Symmetry, 13(2), 310. <https://doi.org/10.3390/sym13020310>