

МОДЕЛЬ АВТОМАТИЗОВАНОГО НАВАНТАЖУВАЛЬНОГО ТЕСТУВАННЯ ПРОГРАМНИХ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ

Романенко А.О., Барковська О.Ю.

Харківський національний університет радіоелектроніки, Харків, Україна

В сучасному світі, де швидкість та надійність програмного забезпечення є критичними факторами успіху, навантажувальне тестування грає важливу роль у впевненості, що програми витримують навантаження реального світу, але більшість сервісів надають можливість проводити таке тестування тільки ручним способом. Окрім того, розробники та користувачі все частіше прибігають до застосування методів штучного інтелекту у різних сферах життя та діяльності. Автоматизація процесу тестування та використання математичного апарату штучних нейронних мереж для рішення цієї задачі не є виключенням. Окремі переваги штучного інтелекту полягають у тому, що він дає змогу тестувальникам програмного забезпечення ефективніше відстежувати та виявляти дефекти, аналізуючи величезні масиви даних, виявляючи шаблони, кореляції та потенційні проблемні області, а також може виконувати тести за кількома сценаріями за короткий час, значно скорочуючи цикл тестування [1]. Серед розглянутих інструментів навантажувального тестування можна виділити Gatling, Locust, BlazeMeter, Artillery, k6, Tsung тощо. Але недоліком наведених інструментів є те, що результат їх роботи залежить від уважності тестувальника, коректності формування множини тестових випадків та неможливості виконувати безперервне тестування.

Метою роботи є розробка моделі автоматизованого навантажувального тестування програмних застосунків, яка, за допомогою методів штучного інтелекту, допоможе автоматизувати роботу інструменту Apache JMeter від Apache Software Foundation, для виконання функціонального, навантажувального тестування та аналізу продуктивності програмних продуктів.

Результатом роботи є досягнення передбачуваної практичної новизни завдяки сумісному використанню генеративної та згорткової нейронної мережі для автоматичного розпізнавання форми реєстрації/авторизації [2], генерування даних та створення сценаріїв для виконання навантажувального тестування, що дозволить підвищити ефективність тестування та зменшити час виконання тестових сценаріїв

Список літератури

1. Goericke S. The future of software quality assurance. – Springer Nature, 2020. – С. 257. DOI: <https://doi.org/10.1007/978-3-030-29509-7>
2. Barkovska O. et al. Adaptation of FPGA architecture for accelerated image preprocessing //Radioelectronic and Computer Systems. – 2023. – №. 2. – С. 94-106. DOI: <https://doi.org/10.32620/reks.2023.2.08>