

ВИКОРИСТАННЯ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ПРИ ТЕСТУВАННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Савчук Тамара, Приймак Наталія

Вінницький національний технічний університет

Анотація

У даній роботі оцінено доцільність використання методів інтелектуального аналізу даних для збільшення ефективності тестування програмного забезпечення. Розглянуто проблеми, що виникають при тестуванні програмних продуктів, а також способи їх розв'язання.

Abstract

This paper evaluated the feasibility of using data mining techniques to increase the efficiency of software testing. Here is also described the problems that arise in testing software and their solutions.

Для забезпечення високої якості програмних продуктів (ПП) до них застосовується тестування, що дозволяє виявити дефекти розробки відповідного програмного забезпечення (ПЗ). Так як складність і різноманітність програм постійно зростає, то їх якість і надійність має важливе значення для споживача. Проаналізувавши наукові праці [1-5], було з'ясовано, що тестування ПП є високовартісною працею. Дослідження, проведене Національним інститутом стандартів і технології встановило, що "щорічна вартість тестування програмного забезпечення оцінюється в діапазоні від 22,2 до 59,5 млрд. доларів США" [6]. Це пояснює необхідність зведення до мінімуму кількості повторного тестування після модифікацій коду програмного забезпечення.

Метою даної роботи є аналіз використання методів інтелектуального аналізу даних (ІАД), що застосовуються для підвищення ефективності тестування ПЗ за рахунок підвищення швидкодії даного процесу (автоматизований вибір тест-кейсів для повторного тестування, прискорення виконання автоматичних тестів), а також прогнозування можливої кількості знайдених дефектів та часу, необхідного на їх виправлення.

Метод міркування на основі прецедентів (case-based reasoning, CBR) [1], основна ідея якого полягає у знаходженні подібних випадків в попередніх ситуаціях, їх аналізу та використання цих знань та інформації до нової проблеми, при тестуванні ПЗ використовується таким чином: після поділу ПП на компоненти, що тестуватимуться, використовується CBR, для знаходження необхідного набору тест-кейсів для конкретного компоненту, базуючись на даних, отриманих при тестуванні схожих компонентів раніше. Такий підхід допомагає скоротити час, необхідний для тестування, оскільки відповідає необхідність у повторному написанні тест-кейсів, а використовуються вже раніше створені.

При тестуванні ПП необхідно розробити набори тестів, за допомогою яких виявляються помилки його розробки. Серед існуючих засобів поширення набули методи інтелектуального аналізу даних для генерації вхідної інформації [2], що ефективно реалізуються з використанням автоматизованого підбору тестів, який скорочує час необхідний для перевірки ПЗ. Уваги заслуговують статистичні методи для генерації початкової інформації, а також асоціативні правила для перевірки відповідності вихідних даних до вхідних.

Дефект програмного забезпечення (баг) – це помилка або збій в комп'ютерній програмі, а їх передбачення – це процес виявлення несправних модулів в програмному забезпеченні. При розробці чи тестуванні ПЗ за допомогою методів класифікації, кластеризації та алгоритмів пошуку асоціативних правил можна виявляти помилки та вирішувати такі задачі [3]:

- групування дефектів відповідно до конкретних компонентів, що дозволить ефективно виправляти баги;
- оцінювання результати перевірки ПЗ інженерами по якості для виявлення необхідності повторної перевірки того чи іншого компонента ПП;
- допомога керівникам проектів у вдосконаленні процесу розробки програмного забезпечення на основі аналізу причин появи схожих дефектів.

Програмне забезпечення із відкритим програмним кодом, також має відкрите сховище дефектів, у яке будь-який користувач може додати знайдену помилку. Автори дослідження, описаного в [4], прогнозують програміста, якого потрібно закріпити за виправленням того чи іншого багу. Дослідники пропонують використовувати алгоритм пошуку асоціативних правил Apriori, щоб передбачити розробника, який працюватиме над виправленням конкретної помилки, враховуючи такі її характеристики: важливість, пріоритет, час необхідний на виконання. Дане прогнозування скорочує час, необхідний для підбору людських ресурсів, потрібних для виправлення багів.

При прогнозуванні часу, необхідного для виправлення певного дефекту, з використанням регресійного аналізу, деякі характеристики багів, які раніше використовувалися для побудови моделей прогнозування часу, необхідного для їх виправлення, не завжди корелюють з цим часом [5]. Результати дослідження показують, що задача пошуку відповідних характеристик є все ще актуальною, оскільки необхідно знайти додаткові атрибути багів для розробки більш загальних моделей прогнозування часу, необхідного для їх виправлення.

Отже, провівши аналіз використання методів ІАД при тестуванні програмного забезпечення, було встановлено, що вони дозволяють підвищити ефективність даного процесу за рахунок збільшення швидкодії тестування, а також спрогнозувати можливу кількість знайдених дефектів та часу, необхідного для їх виправлення.

Список використаних джерел:

1. Ilkhani A., Abaee G. Extracting Test Cases by Using Data Mining; Reducing the Cost of Testing/ A. Ilkhani, G.Abaee// International Journal of Computer Information Systems and Industrial Management Applications. – Shandong, 2011. – Volume 3. – pp. 730 – 737.
2. Zheng W. Automatic Software Testing Via Mining Software Data/ W. Zheng// The Chinese University of Hong Kong September. – Hong Kong, 2011. – p. 274.
3. Yasmeen S. Software Bug Detection Algorithm using Data mining Techniques/ S. Yasmeen// International Journal of Innovative Research in Advanced Engineering (IJIRAE). – India, 2014. – Volume 1. – pp. 134 – 144.
4. Sharma M., Kumari M. Bug Assignee Prediction Using Association Rule Mining/ M. Sharma, M. Kumari// Spriner International Publishing Switzerland. – Cham, 2015. – pp.444 – 457.
5. Bhattacharya P. Bug-fix Time Prediction Models: Can We Do Better?/ P. Bhattacharya// Proceedings of the 8th Working Conference on Mining Software Repositories. – Honolulu, 2011. – pp.207 – 210.
6. National Institute of Standards & Technology. The Economic Impacts of Inadequate Infrastructure for Software Testing, Planning Report 02-3, May 2002.