An Automated Oracle Approach to Test Decision-Making Structures

Seyed Reza Shahamiri, Wan Mohd Nasir Wan Kadir and Suhaimi bin Ibrahim
Department of Software Engineering
Faculty of Computer Science and Information System
Universiti Teknologi Malaysia Skudai, Malaysia
admin@rezanet.com, wnasir@utm.my, suhaimiibrahim@utm.my

Abstract—Decision-making structures are important building blocks in most of the software; however, it may be difficult to verify them because there are various input conditions and several paths causing them to behave differently. Test oracles are reliable sources of how the software must operate. The aim of the present paper is to study the applications of Artificial Neural Networks as an automated oracle to test decision-making structures. First, the decision rules were modeled by the neural network using a training dataset generated based on the software specifications and domain expert knowledge. Next, after the neural network was applied to test a subject-registration application, the proposed approach was evaluated using mutation testing. The accuracy of the resulted oracle is discussed as well.

Keywords—automated software testing; test oracles; decision making structures; artificial neural networks, mutation testing

I. INTRODUCTION

Software testing is the process of finding faults and failures in software products in order to improve their quality and reliability. Since software testing can be very expensive [1], complete testing is practically impossible; thus, testers consider test automation to facilitate the testing process and decrease its costs [8]. Test oracles are reliable sources of how the Software Under Test (SUT) must operate [17]. They are usually required to verify the software behavior in the testing process. Automated test oracles are considered to provide an effective test automation framework. In addition, they should provide correct results for any inputs specified in the software specifications and a comparator to verify the actual behavior automatically.

Decision-making modules are one the important structures that can be seen in most of the software applications. However, it may be difficult to verify them because there are various input conditions and several paths causing them to behave differently. Moreover, providing expected results in nested if-then-else structures requires the programmers and testers to understand the domain of the software completely. Consequently, automated oracles may decrease the complexity and cost of testing such structures significantly.

The aim of the present paper is to study the applications of Artificial Neural Networks (ANNs) as an automated oracle to test decision-making structures. In order to achieve the goal, first the decision-making rules were modeled by the neural network using a training dataset generated based on the software specifications and domain expert knowledge. Next, after applying the trained neural network to test a web-based subject-registration application, the proposed approach was evaluated using mutation testing. The SUT was injected with intentional faults and the ability of the trained ANN (the oracle) to find these mutants was evaluated. A golden version of the case study was implemented to verify the resulted oracle as well.

Before the proposed approach and results of the study being described in detail, the challenges to provide an automated oracle are explained. Then, introducing the ANNs, their applications as automated test oracle to test decision-making structures are illustrated. Finally, the proposed approach and experimental results are described completely.

II. RELATED WORK

Previously ANNs have been successfully applied in software testing. An effective test case selection approach using ANNs is introduced in [2] to study the applications of Input/Output analysis identifying which input attributes mostly affect the value of specific outputs. It showed that I/O analysis can significantly reduce the number of test cases. An ANN was applied to automate I/O analysis to determine important I/O attributes and their ranks. Khoshgoftar et al. [3, 4] proposed a method to employ ANNs predicting the number of faults in SUT based on software metrics. In addition, testability of program modules was studied by the same authors in [5].

ANNs were used as test oracles by Vanmali, Last and Kandel [6]. They modeled an ANN to simulate the software behavior using previous version of the SUT, and applied this model to test unchanged software functionalities in a new version of a small Credit-Approval software application. Using the previous version of the Credit-Approval application, training samples were provided to train the ANN. The main defect of this approach is only unchanged software behaviors can be tested using this approach. Aggarwal, Yogeh, Kaur and Sangwan applied the same approach to solve triangle classification problem [7] and their work was followed by [18].

The approaches mentioned above modeled and tested discrete functions. Mao, Boqin, Li and Yao formulated ANNs as test oracles modeling and testing continuous functions [9]. Consider the continuous function $y=F(x)$, which $x$ is the software input vector, $y$ is corresponding output vector and $F$ is the software behavior. The function $F$