論文の要旨

題　目　A Study on Improvement of Automated Test Input Generation with Machine Learning Techniques in Software Testing
（ソフトウェアテストにおける機械学習技術を用いた自動テスト入力生成の改善に関する研究）

氏　名　　郭　秀景

With the rapid increase of software scale and complexity, the cost of traditional software testing methods will increase faster than the scale of software. In order to improve test efficiency, it is particularly important to automatically generate high-quality test inputs. This article dissertation three approaches focusing on generating test inputs to achieve high test coverage (e.g. branch coverage) and generating test inputs based on boundary value analysis (BVA). We will introduce them successively.

In Chapter 2, we develop a framework for automatic test input generation based on the generative adversarial network (GAN). GAN is employed to train a generative model over execution path information to learn the behavior of the software. Then we can use the trained generative model to produce new test input, and select the test input that can improve the branch coverage according to our proposed selection strategy. Compared to prior work, our proposed method is able to handle programs under test with large-scale branches without analyzing branch expressions. In the experiment, we exhibit the performance of our method by using two modules in GNU Scientific Library. In particular, we consider the application of our method in two testing scenarios; unit testing and integration testing, and conduct a series of experiments to compare the performance of three types of GAN models. Results indicate that the WGAN- GP shows the best performance in our framework. Compared with the random testing method, the WGAN-GP based framework improves the test coverage of five functions out of the seven in the unit testing.

In Chapter 3, we focus on boundary value analysis (BVA). In software testing, a protective measure to prevent faults in the code is to ensure that the behavior on the boundary between the sub-domains of the input space is correct. Therefore, designing test inputs with boundary value analysis (BVA) can detect more errors and improve test efficiency. This dissertation presents an MLP (Multilayer Perceptron) based approach to automatically generate boundary test inputs. Our approach is twofold. First, we train an MLP-based discriminator that determines whether a boundary exists between two test inputs. Second, using the outputs of the discriminator, we create test inputs based on Markov Chain Monte Carlo. We conduct experiments to compare the fault detection capabilities of the MLP-based approach with concolic testing and manually- performed boundary analysis. Results indicate that the MLP-based method outperforms the manually-performed boundary analysis in four of the seven programs tested and concolic testing in three of the seven programs tested.

In Chapter 4, we discuss a boundary coverage metric for BVA. BVA is a common technique in software testing that uses input values that lie at the boundaries where significant changes in behavior are expected. This approach is widely recognized and used as a natural and effective strategy for testing software. Test coverage is one of the criteria to measure how much the software execution paths are covered by the set of test inputs. In this dissertation, we focus on evaluating test coverage with respect to BVA by defining a metric called Boundary Coverage Distance (BCD). The BCD metric measures the extent to which a test set covers the boundaries. In addition, based on BCD, we consider the

optimal test input generation to minimize BCD under the random testing scheme. We propose three algorithms, each representing a different test input generation strategy, and evaluate their fault detection capabilities through experimental validation. The results indicate that the BCD-based approach has the potential to generate boundary values and improve the effectiveness of software testing.

Finally, we summary the contributions of this thesis are; (i) application of the GAN model to automatically generate test inputs, aiming to achieve full branch coverage.(ii) introduce an MLP based approach to automatically generate test inputs with BVA. (iii) propose a novel approach to evaluating test coverage in software testing, focusing on the utilization of Boundary Value Analysis (BVA) and introducing a new metric called Boundary Coverage Distance (BCD).