

Title	プログラム分析における木構造及びグラフ構造へ深層学習の適用
Author(s)	Phan, Anh Viet
Citation	
Issue Date	2018-03
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/15320
Rights	
Description	Supervisor:NGUYEN, Minh Le, 情報科学研究科, 博士

Abstract

The rapid growth of software industry has increased a high demand for tools based on source code analysis to support developers and managers during software development. Source code classifiers are used to organize big projects or a huge amount of open source code on the web, and thus facilitate software reuse and maintenance. With a software defect prediction tool, programmers can easily locate and fix bugs. This leads to an increase in the software quality, and a decrease in the development time and product cost.

Solving software engineering problems is a big challenge. According to previous studies, programming languages contain abundant statistical properties that are difficult to capture by humans. In addition, a program may show different actions in different cases hindering us from discovering its semantic meaning. Although computers can run programs by just executing single instructions, they do not truly understand the programs. For these reasons, although many efforts have made to solve software engineering problems, the achievements are not so high. The traditional approaches build predictive models based on machine learning algorithms and handcrafted features, called software metrics. The drawbacks of such approaches are time-consuming and inaccurate because we must to manually design a set of appropriate metrics and the existing metrics are not enough to capture semantic meanings of programs. Recently, applying deep learning on tree representations to automatically learn programs' features has made a breakthrough in source code analysis. However, such trees simply reflect the program structures and do not reveal the behavior of programs. Thus, tree-based approaches may be inefficient when adapting to several tasks, especially those are relevant to an understanding of semantic meanings like software defect prediction.

In this dissertation, we focus on two main tasks: (1) proposing models and techniques to enhance existing approaches, and (2) formulating a new approach program analysis. For software metrics-based methods, we design a feature weighting model to estimate the importance extent of each metric according to its relevance to class labels. For tree-based approaches, we develop new models as well as refine data by pruning redundant branches to boost the performance. Additionally, we propose a new approach that applies deep learning on assembly code to explore deeper into semantic meanings of programs.

Our contributions can boost the performance of current methods notably and be adapted to various problems of source code analysis.

Keywords: Program Analysis, Abstract Syntax Trees (ASTs), Control Flow Graphs (CFGs), Deep Learning, Convolutional Neural Networks (CNNs).