Chapter 88 Software Defect Prediction Using Genetic Programming and Neural Networks

Mohammed Akour
Yarmouk University, Jordan

Wasen Yahya Melhem
Yarmouk university, Jordan

ABSTRACT

This article describes how classification methods on software defect prediction is widely researched due to the need to increase the software quality and decrease testing efforts. However, findings of past researches done on this issue has not shown any classifier which proves to be superior to the other. Additionally, there is a lack of research that studies the effects and accuracy of genetic programming on software defect prediction. To find solutions for this problem, a comparative software defect prediction experiment between genetic programming and neural networks are performed on four datasets from the NASA Metrics Data repository. Generally, an interesting degree of accuracy is detected, which shows how the metric-based classification is useful. Nevertheless, this article specifies that the application and usage of genetic programming is highly recommended due to the detailed analysis it provides, as well as an important feature in this classification method which allows the viewing of each attributes impact in the dataset.

INTRODUCTION

Software systems are becoming more complicated day after day, consequently the likelihood of defects affecting the system components is expanding. On the other hand, software testing takes up effort, time and budget. The application of data mining such as machine learning approaches to the Software Defect Prediction (SDP) process is a major beneficial and resourceful development. Software experts view SDP as a critical stage where the quality of the software is being tested as it plays a big role in minimizing the

DOI: 10.4018/978-1-7998-0414-7.ch088

claims in software engineering about not being able to find solutions within budget and the time agreed upon. In addition, it provides the customers response concerning the quality of the software which has revealed to be much more satisfactory than before. Countless data mining experts have substituted manual defect prediction methods with a classification model where the software is being classified to being faulty or non-faulty. SDP is considered to be a major part of the testing phase of the SDLC in which testing resources are being efficiently applied with no violations due to the rigorous testing performed (Arora, Ishani, Tetarwal, & Saha, 2015).

Genetic Programming (GP) is one of the evolutionary computation methods that is used for classification and has proven its productivity particularly for dynamic and non-linear classification. In this research paper, four datasets will be studied and classified using GP and Neural Network (NN) to study and experiment the effects of GP and its performance in the classifying procedure and perform a comparison between the NN classification and GP classification. four datasets are collected from Promise Software Engineering Repository, NASA Metric Data Program which are CM1, KC1, KC2, JM1 datasets (Menzies, Tim & Stefano, 2004).

NN is an information-processing model that is stirred by how a human's nervous system performs, where a great number of neurons exist in the brain, this is the fundamental part of the NN model. Neurons are vastly interconnected and perform together to resolve intricate problems. Inputs, weights and bias form the model of a neuron, and there are two types of models: Feed forward and Feed backward NN. The downside of NN is that they are rather problematic to implement due to their complexity (Gayathri, & Sudha, 2014). In order to achieve higher reliability in software and enhance the quality of the software, NN is widely researched and compared with other models and popular in the area of software defect prediction.

The SDP is a vital topic in the field of software engineering with many researches on the issue. Nevertheless, most of the studies have been achieved through design and requirement metrics. The exact and correct estimation of software susceptible to defects can help developers minimize time of testing, costs and enhance the whole procedure of testing by concentrating on classes and functions that are compromised. Software defected datasets have a lot of noise with little faulty classes in comparison with classes that don't contain any bugs. Hence, the classification performance is also minimized due to the imbalanced attributes in the dataset. It is important to note the difference between GA and GP as GP concentrates on the aspect of Classification whereas GA is an optimization method. Correct classifications of faulty software classes can direct test efforts, decrease budgets, progress testing method by concentrating on defective classes and refactoring solutions that are classified as erroneous by building a fault classification model which has shown progression with a mean probability detection of 71% (Menzies et al., 2010).

The OO method unlike old-style programming is built on objects, where information and control are placed separately. OO has become the center of software engineering. This model delivers a novel and enhanced technique to analyze a problem, plan a solution and implement it. It offers better reliability, maintainability and reusability, due to Encapsulation in OOP reusability is accomplished.

Evolutionary algorithms have effectively proven useful in testing software. The use of Object oriented (OO) metrics in predicting defect prone classes were demonstrated by many researchers. Measuring the quality of OO software is not straightforward. Applying genetic algorithm to object-oriented software to uncover defects could be an efficient method.

White, Fan and Oppacher Introduced a model called Basic Object-Oriented GP (OOGP), where object-oriented ideas were added to GP for the purpose of progressing the capacity of GP to handle