Chapter 23
Comparative Analysis of Random Forests with Statistical and Machine Learning Methods in Predicting Fault-Prone Classes

Ruchika Malhotra
Delhi Technological University, India

Arvinder Kaur
GGS Indraprastha University, India

Yogesh Singh
GGS Indraprastha University, India

ABSTRACT

There are available metrics for predicting fault prone classes, which may help software organizations for planning and performing testing activities. This may be possible due to proper allocation of resources on fault prone parts of the design and code of the software. Hence, importance and usefulness of such metrics is understandable, but empirical validation of these metrics is always a great challenge. Random Forest (RF) algorithm has been successfully applied for solving regression and classification problems in many applications. In this work, the authors predict faulty classes/modules using object oriented metrics and static code metrics. This chapter evaluates the capability of RF algorithm and compares its performance with nine statistical and machine learning methods in predicting fault prone software classes. The authors applied RF on six case studies based on open source, commercial software and NASA data sets. The results indicate that the prediction performance of RF is generally better than statistical and machine learning models. Further, the classification of faulty classes/modules using the RF method is better than the other methods in most of the data sets.

DOI: 10.4018/978-1-61350-429-1.ch023

1 INTRODUCTION

As the complexity and the constraints under which the software is developed are increasing, it is difficult to produce software without faults. Such faulty software classes may increase development and maintenance costs due to software failures, and decrease customer's satisfaction (Khoshgoftaar, 2002). Effective prediction of fault-prone software classes/modules may enable software organizations for planning and performing testing by focusing resources on fault-prone parts of the design and code. This may result in significant improvement in software quality (Koru, 2005).

Identification of fault prone classes is commonly achieved through binary prediction models by classifying a class/module as fault-prone or not fault-prone. These prediction models can be built using design metrics, which can be related with faults as independent variables.

RF algorithm is being successfully applied for solving both classification and regression problems. It is therefore important to investigate the capabilities of RF algorithm in predicting software quality. In this work, we investigate the capability of RF algorithm in predicting faulty classes and compare its result with nine other statistical and machine learning methods using WEKA tool (Weka). We investigate the accuracy of the fault proneness predictions using Object-Oriented (OO) design metrics suite given by (Chidamber, 1994) and static code metrics given by Halstead (Halstead, 1977) and McCabe (McCabe, 1976) metrics. In order to perform the analysis we validate the performance of the 10 well known statistical and machine learning methods. We applied RF on two open source software Jedit and Sakura, one NASA data set KC1 (NASA, 2004) and three commercial software (AR1, AR3, AR5) (Promise). The Jedit, sakura and KC1 were developed using OO languages whereas AR1, AR3 and AR5 were developed using procedural language C. The compared models are a) one statistical classifier: Logistic Regression (LR); b) two neural networks: (i) Multi-layer Perceptrons (MLP) and (ii) Radial Basis Function (RBF); c) two Bayesian methods: (i) Bayesian Belief Networks (BBN) and (ii) Naïve Bayes (NB); d) two tree classifiers (i) Random Forest (RF) and (ii) Decision Tree (J48); e) others include (i) Support Vector Machine (SVM), NNage and LogitBoost (LB)..

The contributions of the chapter are summarized as follows: First, both open source software systems and proprietary software are analyzed. Hence we have applied RF on six OO and procedural systems. In previous studies mostly proprietary software were analyzed. Since our analysis is based on six data sets, we will be able to generalize the findings. Second, comparative study of ten different algorithms has been performed to find which algorithm performs the best to predict the fault proneness of the code. The results showed that RF method predict faulty classes with better accuracy.

2 AN OVERVIEW OF RANDOM FOREST (RF) ALGORITHMS

RF combines the advantages of two machine learning methods bagging and random selection. Bagging makes predictions by majority vote of trees by training each tree on bootstrap sample of the training data. Random feature selection (Amit, 1997; Breiman, 2001) searches at each node for the best split over a random subset of the features. Random features and inputs produce good results (Breiman, 2001).

RF uses randomly selected subset of features in order to split at each node while growing a tree. The main characteristics of RF are (Breiman, 2001):

- RF performs faster than boosting and bagging.
- RF is robust to outliers and noise.

Related Content