Chapter 59
Machine Learning Techniques to Predict Software Defect

Ramakanta Mohanty
Keshav Memorial Institute of Technology, India

Vadlamani Ravi
Institute for Development and Research in Banking Technology (IDRBT), India

ABSTRACT

The past 10 years have seen the prediction of software defects proposed by many researchers using various metrics based on measurable aspects of source code entities (e.g. methods, classes, files or modules) and the social structure of software project in an effort to predict the software defects. However, these metrics could not predict very high accuracies in terms of sensitivity, specificity and accuracy. In this chapter, we propose the use of machine learning techniques to predict software defects. The effectiveness of all these techniques is demonstrated on ten datasets taken from literature. Based on an experiment, it is observed that PNN outperformed all other techniques in terms of accuracy and sensitivity in all the software defects datasets followed by CART and Group Method of data handling. We also performed feature selection by t-statistics based approach for selecting feature subsets across different folds for a given technique and followed by the feature subset selection. By taking the most important variables, we invoked the classifiers again and observed that PNN outperformed other classifiers in terms of sensitivity and accuracy. Moreover, the set of 'if- then rules yielded by J48 and CART can be used as an expert system for prediction of software defects.

INTRODUCTION

Machine learning techniques have been dominating in the last two decades. The recently published comprehensive state-of-the-art review (Mohanty et al., 2010) justifies this issue. The ability of software quality models to accurately identify critical faulty components allows for the application of focused verification activities ranging from manual inspection to automated formal analysis methods. Therefore, software quality models to ensure the reliability of the delivered products. Accurate prediction of fault prone modules enables the verification and validation activities that includes quality models: Musa,

DOI: 10.4018/978-1-5225-1759-7.ch059

1998, logistic regression (Basili et al., 1996), discriminant analysis (Khoshgoftaar, 1996), the discriminant power techniques (Schneidewind, 1992), artificial neural network (Khoshgoftaar, 1995), genetic algorithm (Azar et al., 2002), and classification trees (Gokhale et al., 1997; Khoshgoftar et al., 2002; Selby et al., 1988; Fenton et al., 1999).

A wide range of modeling techniques has been proposed and applied for software quality predictions. These include: proposed the Bayesian belief network as the most effective model to predict software quality.

Classification is a popular approach to predict software defects and involves categorizing modules, which is represented by a set of metrics or code attributes into fault prone (fp) non fault prone (nfp) by means of a classification model derived from data (Lessman et al., 2008), statistical methods (Basili et al., 1996; Khoshgoftar & Allen, 1999), tree based methods, (Guo et al., 2004; Khoshgoftar et al., 2000; Menzies et al., 2004; Porter et al., 1990; Selby et al., 1988), neural networks (Khoshgoftar et al., 1995, 1997) and analogy based approaches (El-Emam et al., 2001; Ganeshan et al., 2000; Khoshgoftar et al., 2003), Decision tree (Selby et al., 1988). The discriminative power techniques correctly classified 75 out of 81 fault free modules, and 21 out of 31 faulty modules (Porter et al., 1992). Lessmann et al., (2008) used 10 software development datasets from NASA MDP repository to predict software defects. Most recently, Pendharkar (2010) used the same dataset to test the efficacy of their hybrid exhaustive search and probabilistic neural network (PNN), and simulated annealing (SA) method.

In this chapter, we present a software defect prediction methodology based on GP, BPNN, GMDH, PNN, GRNN, TreeNet, CART, Random Forest Naïve Baye's and J48 on the DATATRIEVE, PC1, PC3, PC4, MC1, KC1, KC2, KC3, CM1 and JM1 datasets.

The rest of the chapter is organized in the following manner. A brief discussion about the overview of machine learning techniques is presented in section 2. Section 3 describes the experimental methodology. Section 4 presents a detailed discussion of the results and discussions. Finally, section 5 concludes the chapter.

OVERVIEW OF THE TECHNIQUES APPLIED

Here we present a brief overview of the machine learning, soft computing and statistical techniques that are employed in this chapter. Since, BPNN is too popular to be overviewed here, the rest of the techniques are presented here.

Group Method of Data Handling (GMDH)

The GMDH was proposed by Ivakhnenko (1968). The main idea behind GMDH is that it tries to build a function (called a polynomial model) that would behave in such a way that the predicted value of the output would be as close as possible to the actual value of output (http://www.inf.kiew.ua/gmdhhome). GMDH (Farlow, 1984) is a heuristic self organizing method that models the input-output relationship of a complex system modeling.

GMDH model with multiple inputs and one output is a subset of the components of the base function in Equation (1) as