Swarm and Evolutionary Computation 32 (2017) 85–109

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Survey Paper

On the application of search-based techniques for software engineering predictive modeling: A systematic review and future directions

Ruchika Malhotra[a,*], Megha Khanna[b,c], Rajeev R. Raje[d]

[a] Department of Software Engineering, Delhi Technological University, India
[b] Delhi Technological University, India
[c] Sri Guru Gobind Singh College of Commerce, University of Delhi, India
[d] Department of Computer and Information Science, Indiana University-Purdue University Indianapolis, Indiana, IN, USA

ARTICLE INFO

Keywords:
Search-based techniques
Change prediction
Defect prediction
Effort estimation
Maintainability prediction
Software quality

ABSTRACT

Software engineering predictive modeling involves construction of models, with the help of software metrics, for estimating quality attributes. Recently, the use of search-based techniques have gained importance as they help the developers and project-managers in the identification of optimal solutions for developing effective prediction models. In this paper, we perform a systematic review of 78 primary studies from January 1992 to December 2015 which analyze the predictive capability of search-based techniques for ascertaining four predominant software quality attributes, i.e., *effort, defect proneness, maintainability* and *change proneness*. The review analyses the effective use and application of search-based techniques by evaluating appropriate specifications of fitness functions, parameter settings, validation methods, accounting for their stochastic natures and the evaluation of developmental models with the use of well-known statistical tests. Furthermore, we compare the effectiveness of different models, developed using the various search-based techniques amongst themselves, and also with the prevalent machine learning techniques used in literature. Although there are very few studies which use search-based techniques for predicting maintainability and change proneness, we found that the results of the application of search-based techniques for effort estimation and defect prediction are encouraging. Hence, this comprehensive study and the associated results will provide guidelines to practitioners and researchers and will enable them to make proper choices for applying the search-based techniques to their specific situations.

1. Introduction

Software engineering predictive modeling (SEPM) helps software professionals in optimizing resources and thus, in producing high quality, efficient, and maintainable software products at low costs. For instance, an efficient predictive model, which suggests the occurrence of defects in a particular class in the forthcoming release of the software, will enable the practitioners to closely evaluate that class by allocating more resources to its testing than the other classes. Such an allocation may help in reducing the errors in the next release of the software. In order to develop an efficient predictor, a classifier needs to be trained with the help of available historical data. The trained classifier can then be used to predict various quality attributes of future software products. Software metrics, which represent various characteristics of a software product, are used for developing and validating these predictive models. Examples of such metrics include Chidamber and Kemerer metrics suite [1], and the Quality Model for Object-Oriented Design metrics suite [2]. These metrics represent various properties of software such as coupling, cohesion, inheritance levels, and size.

A lot of research has been carried out in the field of SEPM, which uses statistical and Machine Learning (ML) techniques for developing predictive models. However, recently a new class of prediction methods, called Search-based techniques, has gained importance in SEPM tasks. A number of studies have established the applicability of the search-based techniques in various software engineering tasks such as requirement analysis, reverse engineering, software product line engineering, software testing, software test case generation, and software project management [3–12]. However, only few studies ascertain the usefulness of the search-based techniques for predicting software development effort, defects, software maintenance effort (maintainability) and maintenance-based changes.

Search-based techniques are meta-heuristic procedures, which are capable of identifying an optimized solution from a large search space

* Corresponding author.
E-mail addresses: ruchikamalhotra2004@yahoo.com (R. Malhotra), meghakhanna86@gmail.com (M. Khanna), rraje@cs.iupui.edu (R.R. Raje).

http://dx.doi.org/10.1016/j.swevo.2016.10.002
Received 24 February 2016; Received in revised form 1 August 2016; Accepted 14 October 2016
Available online 20 October 2016

consisting of potential solutions. They have been successfully used for a number of tasks such as optimal design of FIR filters [13–15], impulse response system identification [16], cloud computing resource scheduling [17] and load forecasting [18]. The search process in a search-based technique is guided by a fitness evaluator which ascertains the appropriateness of a specific solution [11]. The application of search-based techniques for predictive modeling has been advocated by Harman and Jones [19] and Harman [20], as these techniques are efficient in balancing constraints and conflicts. Moreover, they are also efficient in handling noisy, partially inaccurate and incomplete data sets. Other advantages of search-based techniques include their simple problem solving approach and robustness [20,21]. These techniques avoid getting trapped in local optima and conduct the global search efficiently. Harman and Clark [22] have argued that the performance metrics, such as the classification accuracy, can be used by search-based techniques as fitness functions and hence, can be used to create software engineering predictive models.

Given the newly identified relationship between the search-based techniques and predictive modeling, it is necessary to systematically study empirical evidences reported in literature about the use of search-based techniques for the development of software engineering predictive models. There have been a few efforts in that direction in the recent past. For example, a study by Afzal and Torkar [23] focused on a specific search-based technique (Genetic Programming) for estimation and prediction tasks in software engineering. Wen et al. [24] conducted a review of ML techniques for estimating software development effort. Similarly, a study by Ferrucci et al. [12] provides an overview of the application of search-based techniques to the various issues of the software project management. A position paper by Malhotra [25], briefly provides an overview of search-based techniques for defect prediction. Unlike these earlier efforts, this paper presents an extensive analysis about the use of search-based techniques for SEPM, with a specific focus on: (1) software development effort estimation, (2) defect prediction, (3) maintainability prediction, and (4) change prediction. These four attributes are important to estimate in order to effectively allocate resources while developing and maintaining a software product. Moreover, the appropriate prediction of these attributes in the early phases of software development lifecycle helps practitioners to build timely and cost-effective products, while increasing their commitment towards developing a good quality product. We evaluate search-based techniques reported in the literature, covering a time period from January 1992 to December 2015, used for the SEPM tasks. Moreover, this study also explores the effective experimental setups and methods followed in literature for using search-based techniques for SEPM tasks. The four quality attributes evaluated by this study are as follows:

- Software Development Effort/ Effort Estimation: This quality attribute estimates the effort required to develop a specific software product [24]. Effort estimation is important as software project managers require important information from past projects to plan and analyze the effort required for project development [26]. Such knowledge is critical in efficient allocation of human resources so that products can be delivered on time and within the planned budget.
- Defect Prediction: This quality attribute predicts whether a specific module/class of a software will contain defects in the forthcoming releases of the product. Identification of defect prone classes, as indicated earlier, is important as the cost of correcting defects increases exponentially in later phases of the software development cycle [27]. Thus, it is important for researchers to eliminate defects as early as possible to ensure customer satisfaction making the defect proneness prediction an important software quality attribute.
- Maintenance Effort/ Maintainability Prediction: This quality attribute measures the ease with which a software module/class can be modified. It represents the effort required to evolve a particular software module/class. Since, the maintenance phase absorbs 60–70% of the total project resources [27], it is essential for software managers to estimate maintenance effort in the early phases so that proper planning and allocation can be done.
- Change Prediction: This quality attribute predicts whether a specific module/class of software will evolve, i.e., require changes after the software product goes into its operational phase. The determination of change-prone nature of a class helps practitioners in efficient resource allocation as these classes need more attention in the early phases of development so that minimal changes get carried to the next stage. Such steps would ensure an efficient maintainable software product.

It is worth noting that defect prediction and change prediction are binary in nature, i.e., are classification problems. On the other hand, software development effort estimation and maintainability prediction are regression problems.

This paper analyses studies carried out between January 1992 and December 2015. It summarizes and assesses empirical evidences associated with these studies regarding: (1) the use of search-based techniques for software development effort estimation, defect prediction, maintainability prediction and change prediction models; (2) the validation techniques, number of runs to account for the stochastic nature, tools used and performance metrics used for SEPM using search-based techniques; (3) the use of different fitness functions for SEPM; (4) the performance capabilities of the search-based techniques for SEPM; (5) the comparative predictive capabilities of the search-based techniques and the ML techniques; (6) the use of different tests to statistically validate the comparative performance of various techniques, and (7) the advantages and disadvantages of the search-based techniques.

It should be noted that this study is a significant enhancement of author's earlier published article [28], where a limited number of Research Questions (RQs) were evaluated. In this study, we extend the previous work by adding three new research questions (RQ2, RQ3 and RQ6 - see the next section). These questions address the details of the experimental setup (validation method, number of runs, simulation tools and performance metrics) required for SEPM using search-based techniques. We also examine different fitness functions used by search-based techniques to develop effective software engineering predictive models and different statistical tests used in literature to perform comparative evaluation of search-based techniques for SEPM. In addition to the previous quality attributes addressed in [28], we also add a new quality attribute, i.e., maintainability prediction. Moreover, this study also significantly elaborates the discussion of the previous research questions reported in [28].

The rest of the paper is organized as follows: Section 2 presents the method used in conducting the review. Section 3 states the results of the review. Section 4 describes the threats to validity of the review. Section 5 discusses the results of various research questions. Finally, Section 6 presents conclusions of this study and mentions future directions.

2. Method

Kitchenham [29] identified three basic stages in any systematic review - namely, Planning, Conducting and Reporting. Planning is concerned with the establishment of the need of the review and the review protocol. A review protocol, created in the planning stage, clearly defines the research questions to be addressed by the review. A research question for a review study is the aim with which we investigate and review the literature in order to assess its current state. These research questions should be generated with clear objectives in a researcher's mind so as to guide the research methodology and the investigation process. Research questions are based on the intent with which we explore the studies for analyzing a specific topic