

Guest editorial: special issue on predictive models for software quality

Leandro L. Minku¹ · Ayşe B. Bener² · Burak Turhan³

Published online: 16 April 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Software systems are increasingly large and complex, making activities related to ensuring software quality increasingly difficult. In this context, techniques able to automatically retrieve knowledge from software data in order to improve software quality are highly desirable. Predictive modelling has been showing promising results in this area. For instance, it can be used to learn the relationship between features retrieved from software processes, software usage or software itself and certain properties of interest, e.g., the presence of bugs, the likelihood of changes leading to crashes and the presence of code smells. Such knowledge can be particularly useful to improve the quality of large and complex systems.

With this in mind, this special issue aims at investigating predictive models for software quality. We solicited submissions that provide an in depth understanding of when, why and how algorithms to create predictive models work in the context of software quality. We believe that such understanding will greatly benefit the software quality community, given that it will improve the external validity of studies and provide insights into how to improve algorithms further.

Following an open call for papers, the special issue received a total of 11 submissions, one of which was withdrawn. The remaining 10 submissions were peer-reviewed by experts in the field. At the end, four papers were selected for inclusion in this special issue:

✉ Leandro L. Minku
leandro.minku@leicester.ac.uk

Ayşe B. Bener
ayse.bener@ryerson.ca

Burak Turhan
burak.turhan@brunel.ac.uk

¹ Department of Informatics, University of Leicester, University Road, Leicester, LE1 7RH, UK

² Department of Mechanical and Industrial Engineering, Ryerson University, 350 Victoria Street, Toronto, Ontario, M5B 2K3, Canada

³ Department of Computer Science, Brunel University London, Kingston Lane, Uxbridge, UB8 3PH, UK

‘Software Defect Prediction: Do Different Classifiers Find the Same Defects?’ by David Bowes, Tracy Hall and Jean Petrić presents an insightful study of Random Forest, Naïve Bayes, RPart and Support Vector Machine classifiers for within-project software defect prediction, based on open source and industrial data sets. The study goes beyond a mere comparison of predictive performance to show that different classifiers complement each other even though they have similar predictive performance as they detect different sets of defects. Combining different classifiers in ensembles is thus a recommended strategy for software defect prediction. The results also suggest that the typical majority vote mechanism used to combine classifiers in ensembles may not be ideal for software defect prediction. Therefore, other combination mechanisms should be investigated.

‘An Empirical Study of Crash-inducing Commits in Mozilla Firefox’ by Le An, Foutse Khomh and Yann-Gaël Guéhéneuc performs an analysis of crash-inducing commits in Mozilla Firefox, with the aim of preventing future crashes. Four classifiers (Generalized Linear Model, Naïve Bayes, Decision Tree and Random Forest) are built to predict crash-prone bugs as soon as their code is committed in order to avoid crashes in the production environment. The study also reveals certain characteristics associated with crash-prone commits in Mozilla Firefox, such as the level of expertise of the developers committing the code, the number of lines of code and the effort required to fix the bugs.

‘Stability prediction of the software requirements specification’ by José del Sagrado and Isabel María del Águila proposes to use Bayesian Networks to gain insights into whether requirements specifications are stable or need to be revised. In their study, the authors manually built the structure of a network to predict requirements’ stability with the aid of two software engineers. The network was then integrated with a Computer-Aided Requirements Engineering tool and validated based on a case study with a large-scale real world data set.

‘Towards improving decision making & estimating the value of decisions in value-based software engineering: The VALUE Framework’ by Emilia Mendes, Pilar Rodriguez, Vitor Freitas, Simon Baker and Amine Atoui presents a new framework called VALUE to support value-based decision-making for software-intensive projects and services development. The framework elicits stakeholders’ knowledge regarding factors taken into account for decision-making in a company. These factors are used as input to a tool to support decision-making and to create the structure of a Bayesian Network for estimating the value of decisions. Data on decisions made by the stakeholders in the past is collected with the tool and used to train the Bayesian Network, to estimate the value of future decisions in that company. The authors conducted an empirical study at a company to show the viability of their proposed framework.

Collectively, the four accepted papers show that the area of predictive models for software quality is not only expanding to incorporate new topics of interest, but also becoming more mature in terms of providing a more thorough understanding of the issues surrounding existing topics.

The special issue was preceded by the 11th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE), held in Beijing, China, on 21 October 2015. The articles have undergone rigorous peer-review according to the journal’s high standards. We would like to extend our sincere thanks to the authors for their contributions, to reviewers for their invaluable assistance and to the Editor-in-Chief for making this special issue possible. We hope that you will enjoy reading these interesting contributions.

Guest Editors

Leandro L. Minku, University of Leicester, UK

Ayşe B. Bener, Ryerson University, Canada

Burak Turhan, Brunel University London, UK