Contribution ID: **63** Type: **Oral Presentation**

A comparative analysis of machine learning techniques for prediction of software change: a HEP software case study (Remote Presentation)

Tuesday, 25 August 2020 17:00 (30 minutes)

The objective of this work is to assess various machine learning techniques in order to predict change proneness of software modules [1, 2] that are used in High Energy Physics (HEP) domain. This type of prediction determines the modules that have a higher probability of modification across the version of a software. Therefore, together with other software quality assessment tools, it can help software developers during the software life cycle, leading them to focus on the modules that need the most attention.
The present study has been carried out by measuring a set of software metrics for some HEP software, notably Geant4 [3] and ROOT [4], making use of proper software metrics tools, both open source and proprietary. Taking into account the existing documentation, changes to the code made during the development and maintenance phases of the software (such as improvements, fixed bugs and warnings) have been traced and codified. The resultant datasets have then been used to identify software metrics suitable to predict change proneness of software modules.

In our previous studies [4, 5], work has been performed on unlabelled software metrics datasets, whose values belong to (HEP and non-HEP) software modules. These modules have been characterized by the lack of some kind of information (like their defectiveness) that can help developers to determine their quality. This lack is mainly due to the fact that the measurement of software metrics has been done on ongoing software projects with a partial knowledge of software changes. Therefore, our previous work has been initially focused on software modules defectiveness prediction by using unsupervised machine learning techniques. In the present work, our activity aims at investigating code changes by applying both supervised and semi-supervised machine learning techniques.

The application of machine learning techniques to denote problems in HEP software is an open area of research: software metrics datasets are either incomplete or absent. To tackle this problem, a new dictionary of software changes has been defined by leveraging the collected information related to software changes. The dictionary is essential in the classification phase of the various software modules: terms like warning, fixed bug, minor fix and optimization have been opportunely codified and used to label each module according to the correspondent types of changes (e.g., a module could include minor fix and optimization changes related to an inappropriate development; another module could include warning and performance changes). The defined software changes dictionary will derive by the available software documentation of the considered HEP software. This work will in future be extended to generalize our achievements also to non-HEP software.

As a final remark, the relationship between software metrics and code changes have been investigated by applying existing machine learning techniques. These techniques are compared
in terms of performance indicators such as F-measure and Kappa statistics. Our labelling approach can be easily automated and used by software scientists to monitor their software over time and identify the modules that require a particular attention.

[1] F. Khomh, M. Di Penta, Y.-G. Guéhéneuc, and G. Antoniol, 2012. "An exploratory study of the impact of antipatterns on class change-and fault-proneness", Empirical Softw. Engg. 17, 3, 243–275, https://doi.org/10.1007/s10664-011-9171-y
[2] Q. D. Soetens, S. Demeyer, A. Zaidman, and J. Perez, 2016. "Change-based test selection: An empirical evaluation", Empirical Softw. Engg. 21, 5, 1990–2032, https://doi.org/10.1007/s10664-015-9405-5
[3] E. Ronchieri, M. G. Pia, T. Basaglia, and M. Canaparo, 2016. "Geant4 Maintainability Assessed with Respect to Software Engineering References", In Proc. of IEEE NSS/MIC/RTSD 2016, https://doi.org/10.1109/NSSMIC.2016.8069636
[4] E. Ronchieri, M. Canaparo, D. C. Duma, and A. Costantini, 2019. "Data mining techniques for software quality prediction: a comparative study", In Proc. of IEEE NSS MIC 2018, https://doi.org/10.1109/NSSMIC.2018.8824313
[5] M. Canaparo, and E. Ronchieri, 2019. "Data mining techniques for software quality prediction in open source software: an initial assessment", In Proc. of CHEP 2018, EPJ Web of Conferences 214, 05007, https://doi.org/10.1051/epjconf/201921
[6] E. Ronchieri, M. Canaparo, and D. Salomoni, Machine Learning Techniques for Software Analysis of Unlabelled Program Modules, under review of Proc. of ISGC 2019

Primary authors: Dr COSTANTINI, Alessandro (INFN-CNAF); Dr SALOMONI, Davide (INFN); DUMA, Doina Cristina (INFN - CNAF); Dr RONCHIERI, Elisabetta (INFN CNAF); Mr CANAPARO, Marco (INFN)

Presenter: Dr RONCHIERI, Elisabetta (INFN CNAF)

Session Classification: Physics & Engineering

Track Classification: Physics (including HEP) and Engineering Applications