Contribution ID: **70** Type: **Oral Presentation**

Machine Learning Techniques for Software Analysis of Unlabelled Program Modules

Friday, 5 April 2019 11:20 (30 minutes)

Software analysis is of vital importance in the assessment of software characteristics. It is usually based on software measurement, defect data and techniques derived from both statistics and machine learning.

Machine learning has been widely adopted in the field of Software Engineering (SE). For typical SE tasks, machine learning helps computer engineers e.g. extract requirements from natural language text [1], generate source code [2] and predict defects in software [3]. To accomplish these tasks, data have to be collected and properly preprocessed before the application of machine learning techniques. Typical data preprocessing operations may include replacement of missing values and/or removal of inconsistencies. The obtained datasets are therefore composed of a set of instances (i.e. modules, such as files, classes and functions) and features (i.e. software metrics and defect data) and are used to train software quality models by using supervised learning approaches. In SE practice, software dataset may lack some information such as the software module defectiveness, which is mandatory for the application of supervised learning techniques.

Typical datasets employed in machine learning, known as labelled datasets, are related to a single software project whose features have been extracted over time. Among features, defect data may be difficult to collect especially when dealing with new projects or projects with partial historical data. The datasets without defect data are known as unlabelled and represent the majority of software datasets: the production of a labelled software dataset requires effort and time, penalizing a real application of machine learning techniques to predict modules defectiveness. Only in the last decade the unlabelled datasets have been explored with the purpose of conducting analysis and predictions [4, 5].

Machine learning problems have been increasing in complexity over the years leading to a greater resources' consumption. Therefore, average systems and platforms are not considered suitable for performing machine learning algorithms: the number of permutations requested to perform a prediction analysis is extremely time consuming. Cloud computing services have given the chance to overcome these limitations by giving access to large-scale computing and storage resources with little effort.

In this study, we are going to present the analysis of existing unlabelled datasets by implementing models in different available frameworks, such as TensorFlow, Theano and Keras [6], and running in Python. We have evaluated these frameworks on considering three aspects: extensibility, hardware utilization and speed. For this work, we have exploited (also GPU-equipped) cloud computing infrastructure to determine the best models according to common intrinsic evaluation metrics.

Due to the lack of a comprehensive study about practical aspects of software analytic models, we aim at providing a procedure to perform software defect prediction in the scientific environment in order to minimize human effort. Furthermore, we intend to reduce the distance between theory and practice by providing strengths and limitations of the considered frameworks to enable users to assess suitability according to their requirements.

[1] Madala, K., Gaither, D., Nielsen, R., & Do, H. "Automated identification of component state transition model elements from requirements," in International Requirements Engineering Conference Workshops, pp.386-392, 2017.

[2] Joulin, A., & Mikolov, T. "Inferring algorithmic patterns with stack-augmented recurrent nets," NIPS'15, pp. 190-198, 2015.

[3] Tong, H., Liu, B., & Wang, S. "Software defect prediction using stacked denoising autoencoders and two-stage ensemble learning," IST'17, 2017.

[4] Catal, C., Sevim, U., & Diri, B. "Clustering and metrics thresholds based software fault prediction of unlabeled program modules," in Information Technology: New Generations, ITNG '09, Sixth International Conference on, pp. 199–204, April 2009.

[5] Zhong, S., Khoshgoftaar, T., & Seliya, N. "Unsupervised learning for expert-based software quality estimation," in High Assurance Systems Engineering, Proceedings. Eighth IEEE International Symposium on, pp. 149–155, March 2004.

[6] Bahrampour, S., Ramakrishnan,N., Schott, L., & Shah, M. “Comparative Study of Deep Learning Software Frameworks,” https://arxiv.org/pdf/1511.06435.pdf

Primary author: Dr RONCHIERI, Elisabetta (INFN CNAF)

Co-authors: Dr SALOMONI, Davide (INFN); Mr CANAPARO, Marco (INFN)

Presenters: Dr SALOMONI, Davide (INFN); Dr RONCHIERI, Elisabetta (INFN CNAF); Mr CANAPARO, Marco (INFN)

Session Classification: Physics & Engineering Application

Track Classification: Physics (including HEP) and Engineering Applications