A Comparative Study to Benchmark Cross-project Defect Prediction Approaches

Steffen Herbold, Alexander Trautsch, Jens Grabowski
University of Goettingen, Insititute of Computer Science
Göttingen, Germany
{herbold,alexander.trautsch,grabowski}@cs.uni-goettingen.de

EXTENDED ABSTRACT

Cross-Project Defect Prediction (CPDP) as a means to focus quality assurance of software projects was under heavy investigation in recent years. However, within the current state-of-the-art it is unclear which of the many proposals performs best due to a lack of replication of results and diverse experiment setups that utilize different performance metrics and are based on different underlying data. Within this article [2, 3], we provide a benchmark for CPDP. Our benchmark replicates 24 CPDP approaches proposed by researchers between 2008 and 2015. Through our benchmark, we answer the following research questions:

- **RQ1:** Which CPDP approaches perform best in terms of *F-measure*, *G-measure*, *AUC*, and *MCC*?
- **RQ2:** Does any CPDP approach consistently fulfill the performance criteria for successful predictions postulated by Zimmermann *et al.* [4], i.e., have at least 0.75 *recall*, 0.75 *precision*, and 0.75 *accuracy*?
- **RQ3:** What is the impact of using only larger products (> 100 instances) with a certain balance (at least 5% defective instances and at least 5% non-defective instances) on the benchmark results?
- **RQ4:** What is the impact of using a relatively small subset of a larger data set on the benchmark results?

We identified 5 public data sets, which contain defect data about 86 software products that we used to answer these research question. The advantage of using multiple data sets was that we could increase the number of software products and, thereby, increase the external validity of our results. Moreover, we wanted to use multiple performance criteria for the evaluation of the CPDP approaches. Therefore, RQ1 ranks approaches not just using a single criterion, but using the four performance metrics *AUC*, *F-measure*, *G-measure*, and *MCC*. Existing approaches for the ranking of statistically different approaches neither account for software products from different data sets, nor multiple performance metrics. Therefore, we defined a new approach for the combination of separate rankings for the performance criteria and data sets, into one common ranking.

Figure 1 depicts the results for RQ1. The results show that an approach proposed by Camargo Cruz and Ochimizu [1] performs best and even outperforms cross-validation. Moreover, our results show

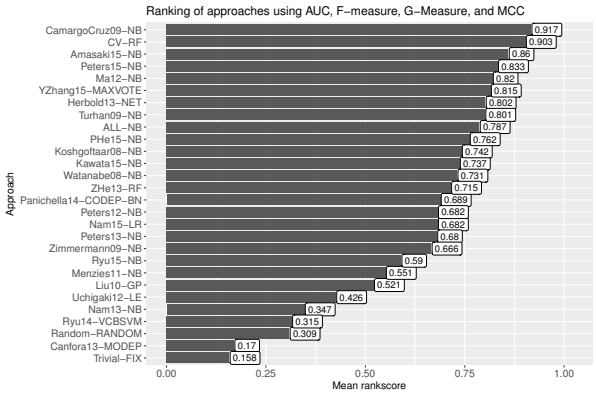

Figure 1: Mean rank score over all data sets for the metrics *AUC*, *F-measure*, *G-measure*, and *MCC*. In case multiple classifiers were used, we list only the result achieved with the best classifier.

that only 6 of the 24 approaches outperform one of our baselines, i.e., using all data for training without any transfer learning. Regarding RQ2, we determined that predictions only seldomly achieve a high performance of 0.75 *recall*, *precision*, and *accuracy*. The best CPDP approaches only fulfill the criterion for 4 of the 86 products, i.e., 4.6% of the time. Thus, CPDP still has not reached a point where the performance of the results is sufficient for the application in practice.

RQ3 and RQ4 were used to see if results are affected by subsetting data, as is often done for defect prediction experiments. For RQ3, i.e., using a large subset, we determined no difference between using all data and using the subset. For RQ4, i.e., using a small subset of of data, we found that there are statistically signifcant differences in reported performances of up to 5%. Thus, the use of small subsets should be avoided.

ICSE '18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5638-1/18/05.
https://doi.org/10.1145/3180155.3182542

REFERENCES

[1] A. E. Camargo Cruz and K. Ochimizu. 2009. Towards logistic regression models for predicting fault-prone code across software projects. In *Proc. 3rd Int. Symp. on Empirical Softw. Eng. and Measurement (ESEM)*. IEEE Computer Society. https://doi.org/10.1109/ESEM.2009.5316002

[2] S. Herbold, A. Trautsch, and J. Grabowski. 2017. A Comparative Study to Benchmark Cross-project Defect Prediction Approaches. *IEEE Trans. Softw. Eng.* Online First (2017). https://doi.org/10.1109/TSE.2017.2724538

[3] S. Herbold, A. Trautsch, and J. Grabowski. 2017. Correction of "A Comparative Study to Benchmark Cross-project Defect Prediction". *CoRR* abs/1707.09281 (2017). https://arxiv.org/abs/1707.09281

[4] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy. 2009. Cross-project defect prediction: a large scale experiment on data vs. domain vs. process. In *Proc. the 7th Joint Meet. Eur. Softw. Eng. Conf. (ESEC) and the ACM SIGSOFT Symp. Found. Softw. Eng. (FSE)*. ACM. https://doi.org/10.1145/1595696.1595713