# Paper Six Summary

Matthew Neal, Joseph Sankar, and Alexander Sobran

October 27, 2015

## Reference

Rahman et al. [2] listed below.

## Important Keywords

**Cross-project defect prediction** Using data from one project to predict defects in another.

**Within-project defect prediction** Using data from previous releases of a project to predict defects in future releases. For new projects, the lack of historical defect data makes this kind of defect prediction almost impossible.

**F-measure** The harmonic mean of precision and recall. A unified score used to balance the trade-off between precision and recall.

**Over-fitting** When a model has completely learned all the variances of the training data and has lost significant generality leading to significantly worse predictive performance on unseen test data. This property usually manifests as very low training error and very high test error. Over-fitting becomes more probable as a model gains greater complexity.

## Feature Extraction

**Motivational Statements** While within-project defect prediction can be very effective, new projects don't have the volume of data needed to create these models. Cross-project defect prediction models aim to help with this issue, but so far the results have largely been disappointing. The authors hope to show that cross-project defect prediction can be roughly as effective as traditional defect prediction by using a different set of measures, namely those based on a variety of tradeoffs of time-and-cost vs. quality.

**Hypothesis** The hypothesis of this paper is that performance of cross-project defect prediction models is similar to the performance of within-project defect prediction when assessing performance within reasonable and practical resource limitations.

**Statistical tests** The authors used the area under the ROC curve (AUC), AUCCE, precision, recall, and f-measure statistical tests to report their results. They varied the cutoffs for precision, recall, and f-measure but used the standard 0.5 cutoff for their sanity checks.

**Informative Visualization** Figure 2 gives an informative comparison of cross and within-project model performance assessed by multiple performance metrics. It visualizes how well the distribution of results overlap for within and cross-project model performance even though within-project generally has higher performance. The addition of p-values shows how the differences lack significance. Figure 4 is also very informative, showing how weighting the defects by density affects AUC. Both cross-project and within-project have similar results and the addition of random and optimal baselines are useful for comparison.

## Possible Improvements

- In the Results section, some of the graphs are hard to read because they have been scaled down to fit on the page. Look at Figures 5 and 7 for example. The y-axis label runs into the labels along the y-axis. To fix this, the individual graphs should be split so that there is only one per column, or they should be combined and made bigger.

## Connection to Other Papers

The authors state that they are examining code at the file level. They reference our first paper on ecological inference [1] when stating that they feared that a model on a coarser level of aggregation may not be a good predictor at the file level.

## References

[1] Daryl Posnett, Vladimir Filkov, and Premkumar Devanbu. Ecological inference in empirical software engineering. In *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, ASE '11, pages 362–371, Washington, DC, USA, 2011. IEEE Computer Society.

[2] Foyzur Rahman, Daryl Posnett, and Premkumar Devanbu. Recalling the "imprecision" of cross-project defect prediction. In *Proceedings of the ACM*

*SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, FSE ’12, pages 61:1–61:11, New York, NY, USA, 2012. ACM.