Towards a Better Understanding of the Impact of Experimental Components on Defect Prediction Modelling

Chakkrit Tantithamthavorn[1]
[1]Nara Institute of Science and Technology, Japan. [2]Queen's University, Canada.
Adviced by: Kenichi Matsumoto[1] and Ahmed E. Hassan[2]
{chakkrit-t,matumoto}@is.naist.jp, ahmed@cs.queensu.ca
Homepage: http://chakkrit.com

ABSTRACT

Defect prediction models are used to pinpoint risky software modules and understand past pitfalls that lead to defective modules. The predictions and insights that are derived from defect prediction models may not be accurate and reliable if researchers do not consider the impact of experimental components (e.g., datasets, metrics, and classifiers) of defect prediction modelling. Therefore, a lack of awareness and practical guidelines from previous research can lead to invalid predictions and unreliable insights. In this thesis, we investigate the impact that experimental components have on the predictions and insights of defect prediction models. Through case studies of systems that span both proprietary and open-source domains, we find that (1) noise in defect datasets; (2) parameter settings of classification techniques; and (3) model validation techniques have a large impact on the predictions and insights of defect prediction models, suggesting that researchers should carefully select experimental components in order to produce more accurate and reliable defect prediction models.

CCS Concepts

•General and reference → *Experimentation;* **•Software and its engineering** → **Software defect analysis;**

Keywords

Defect prediction modelling, experimental components.

1. INTRODUCTION

Defect models, which identify defect-prone software modules using a variety of software metrics, serve two main purposes. First, defect models can be used to *predict* modules that are likely to be defect-prone. Software Quality Assurance (SQA) teams can use defect models in a prediction setting to effectively allocate their limited resources to the modules that are most likely to be defective. Second, defect models can be used to *understand* the impact that various software metrics have on the defect-proneness of a module. The insights derived from defect models can help software teams to avoid past pitfalls that lead to defective modules.

The predictions and insights that are derived from defect prediction models may not be accurate and reliable if researchers do not consider the impact that experimental components (e.g., datasets, metrics, and classifiers) of defect prediction modelling. Indeed, there exists a plethora of research that raise concerns about the impact of experimental components on defect prediction modelling [8, 11]. For example, Shepperd *et al.* [11] find that the reported performance of a defect prediction model shares a strong relationship with the group of researchers who construct the models. Their observations suggest that many published defect prediction studies are biased, and calls their validity into question.

To assess the impact of experimental components on defect prediction modelling, we investigate the association between the reported performance of a defect model and the used experimental components (i.e., datasets, metrics, and classifiers). Through a case study of 42 primary defect prediction studies [14], we find that the experimental components (i.e., metrics) that are used to construct defect prediction models share a stronger relationship with the reported performance than research group does, suggesting that experimental components of defect prediction modelling may impact the conclusions of defect prediction studies.

In this thesis, we investigate the impact that (1) noise in defect datasets and (2) parameter settings of classification techniques have on the predictions and insights of defect prediction models. In addition, defect prediction models may produce an unrealistic estimation of model performance when inaccurate and unreliable model validation techniques are applied, which could lead to incorrect model selection in practice and unstable conclusions of defect prediction studies. Thus, we further investigate the impact that (3) model validation techniques have on the accuracy and reliability of performance estimates that are produced by defect prediction models. Through case studies of systems that span both proprietary and open-source domains, we demonstrate that these three experimental components have a large impact on the predictions and insights of defect prediction models, suggesting that researchers should carefully select experimental components in order to produce more accurate and reliable defect prediction models.

Section 2 illustrates the relevance of experimental components for defect prediction modelling. Section 3 presents our thesis statement, while Section 4 summarizes the current state of the work. Section 5 derives practical guidelines for future research. Section 6 draws conclusions and contributions. Finally, Section 7 describes the progress and outlook.

ICSE '16 Companion, May 14-22, 2016, Austin, TX, USA
© 2016 ACM. ISBN 978-1-4503-4205-6/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2889160.2889256

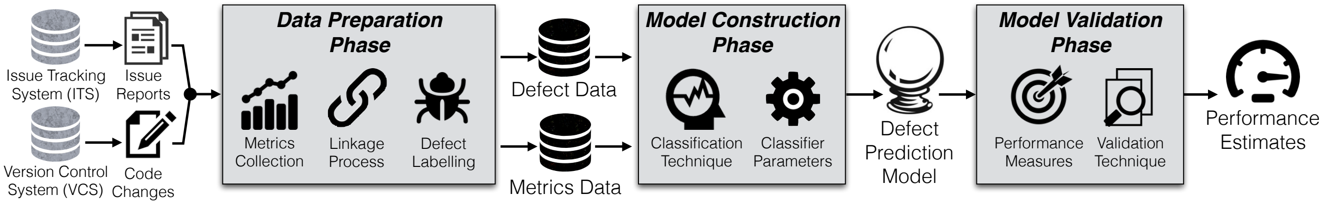

Figure 1: An overview of experimental components of defect prediction modelling.

2. THE RELEVANCE OF EXPERIMENTAL COMPONENTS FOR DEFECT PREDICTION MODELLING

Shepperd *et al.* [11] find that the reported performance of a defect prediction model shares a strong relationship with the group of researchers who construct the models. This observation raises several concerns about the state of the defect prediction field. We suspect that research groups are likely to reuse experimental components (e.g., datasets, metrics, and classifiers) across their various studies. This tendency to reuse experimental components would introduce a strong association among the explanatory variables of Shepperd *et al.*, i.e., research group and the experimental components (i.e., dataset family, metric family, and classifier family).

To this end, we investigate the association between the reported performance of a defect prediction model and the explanatory variables of Shepperd *et al.*'s data. Through a case study of 42 primary defect prediction studies [14], we find that (1) research group shares a strong association with the dataset and metrics families that are used in building models; (2) the strong association among explanatory variables introduces interference when interpreting the impact that research group has on the reported model performance; and (3) after mitigating the interference, we find that the experimental components (i.e., metric family) that are used to construct defect prediction models share a stronger relationship with the reported performance than research group does. Our findings suggest that experimental components of defect prediction modelling may influence the conclusions of defect prediction studies.

Figure 1 provides an overview of the defect prediction modelling. To develop a defect prediction modelling of Figure 1, we first need to *prepare a metrics dataset* of software modules (e.g., size, complexity, process metrics), which are typically collected from a version control system (VCS). Second, we need to *prepare a linked dataset* of issue reports (i.e., a report described defects or feature requests) and code changes to identify which modules are changed to address an issue report. Third, we need to *label defective modules* if they have been affected by a code change that addresses an issue report that is classified as a defect. Forth, we *train defect models* using a machine learning technique. Fifth, we need to *configure parameter settings* of machine learning techniques that control their characteristics (e.g., the number of trees in a random forest classifier). Sixth, we *select performance measures* to quantify the performance of defect prediction models. Finally, we *validate the models* in order to estimate the performance of the model when it is applied to new software modules and *interpret the models* in order to understand past pitfalls that lead to defective modules.

Various issues can arise when constructing defect prediction models. Indeed, there exists a plethora of research that raise concerns about the impact of experimental components on defect prediction modelling. We discuss below the concerns with respect to prior works.

2.1 Data Preparation

The process of selecting module metrics can impact the performance of defect prediction models [10]. Moreover, the process of linking issue reports with code changes can generate noise in defect datasets, since the linkage process often depends on manually-entered links that are provided by developers [1]. Even if all of the links between issue reports and code changes are correctly recovered, noise may creep into defect datasets if the issue reports themselves are mislabelled [4], i.e., issue reports that describe defects but were not classified as such (or vice versa). Yet, little is known about the characteristics of noise that is generated by issue report mislabelling and its impact on the predictions and insights derived from defect prediction models.

2.2 Model Construction

To construct defect prediction models, prior work has explored the use of various classification techniques in order to train defect prediction models. Recent research finds that the choice of classification techniques has a large impact on the performance of defect prediction models [2]. Such classification techniques often have *configurable parameters* that control their characteristics (e.g., the number of trees in a random forest classifier). Yet, little is known about the impact that the choice of configurable parameter settings can have on the performance of defect prediction models.

2.3 Model Validation

The performance of defect prediction models can be quantified using a variety of threshold-dependent (e.g., precision, recall) and threshold-independent (e.g, Area Under the the receiver operating characteristic Curve (AUC)) performance measures. To validate defect prediction models, model validation techniques (e.g., k-fold cross-validation) are commonly used to estimate how well a model will perform on unseen data. Recent research has raised concerns about the unrealistic performance estimation that are produced by model validation techniques when they are applied to defect prediction models [9]. Yet, little is known about how accurate and reliable the performance estimates of model validation techniques tend to be.

3. THESIS STATEMENT

The empirical evidence in Section 2 leads us to the formation of our thesis statement, which we state as follows.

A lack of awareness of experimental components can interfere with defect prediction models, which can lead to invalid predictions and unreliable insights. With more mindful experimental component selection, the predictions and insights that are derived from defect prediction models will be more accurate and reliable.

Our thesis statement will be addressed by three major experimental components in defect prediction modelling:

Study (1) **Overlooking noise generated by issue report mislabelling.** We investigate the impact that realistic noise generated by issue report mislabelling has on the predictions and insights of defect prediction models.

Study (2) **Overlooking the optimal parameter settings of classification techniques.** We investigate the impact that parameter settings of classification techniques have on the accuracy and reliability of the performance of defect prediction models when automated parameter optimization is applied.

Study (3) **Overlooking the most accurate and reliable model validation techniques.** We investigate the impact that model validation techniques have on the accuracy and reliability of the performance of defect prediction models.

4. CURRENT STATE OF THE WORK

4.1 Overlooking noise generated by issue report mislabelling

Motivation. The accuracy and reliability of a prediction model depends on the quality of the data from which it was trained. Therefore, defect prediction models may be inaccurate and unreliable if they are trained using noisy data [4, 5]. Recent research shows that noise that is generated by *issue report mislabelling*, i.e., issue reports that describe defects but were not classified as such (or vice versa), may impact the performance of defect models [5]. Yet, while issue report mislabelling is likely influenced by characteristics of the issue itself — e.g., novice developers may be more likely to mislabel an issue than an experienced developer — the prior work randomly generates mislabelled issues.

Approach. We investigate whether mislabelled issue reports can be accurately explained using characteristics of the issue reports themselves, and what is the impact that a realistic amount of noise has on the predictions and insights derived from defect models. Using the manually-curated dataset of mislabelled issue reports provided by Herzig *et al.* [4], we generate three types of defect datasets: (1) *realistic noisy* datasets that contain mislabelled issue reports as classified manually by Herzig *et al.*, (2) *random noisy* datasets that contain the same proportion of mislabelled issue reports as contained in the realistic noisy dataset, however the mislabelled issue reports are selected at random, and (3) *clean* datasets that contain no mislabelled issues.

Results. We find that (1) issue report mislabelling is not random; (2) precision is rarely impacted by mislabelled issue reports; (3) however, models trained on noisy data typically achieve 56%-68% of the recall of models trained on clean data; and (4) only the metrics in top influence rank of our defect models are robust to the noise introduced by mislabelling. More details are provided in our publication [12].

4.2 Overlooking the parameters of classification techniques

Motivation. Defect prediction models are classifiers that are trained to identify defect-prone software modules. Such classifiers have *configurable parameters* that control their characteristics (e.g., the number of trees in a random forest classifier). Recent studies show that these classifiers may underperform due to the use of suboptimal default parameter settings [3]. However, it is impractical to assess all of the possible settings in the parameter spaces.

Approach. We perform a literature analysis that reveals that 26 of the 30 most commonly-used classification techniques (87%) require at least one parameter setting. Since such parameter settings may impact the performance of defect prediction models, the settings should be carefully selected. We then investigate the improvement and the reliability of the performance of defect prediction models when Caret [6] — an off-the-shelf automated parameter optimization technique — is applied. Caret evaluates candidate parameter settings and suggests the optimized setting that achieves the highest performance.

Results. We find that (1) Caret improves the AUC performance of defect prediction models by up to 40 percentage points; and (2) Caret-optimized classifiers are at least as reliable as classifiers that are trained using the default settings. Our results lead us to conclude that parameter settings can indeed have a large impact on the performance of defect prediction models, suggesting that researchers should experiment with the parameters of the classification techniques. More details are provided in our publication [15].

4.3 Overlooking the most accurate and reliable model validation techniques

Motivation. Prediction models may provide an unrealistically optimistic estimation of model performance when (re)applied to the same sample with which that they were trained. To address this problem, *Model Validation Techniques (MVTs)* (e.g., k-fold cross-validation) are used to estimate how well a model will perform on unseen data [7]. Recent research has raised concerns about the *accuracy* (i.e., how much do the performance estimates differ from the ground truth?) and *reliability* (i.e., how much do performance estimates vary when the experiment is repeated?) of model validation techniques when applied to defect prediction models [9]. An optimal MVT would not overestimate or underestimate the ground truth performance. Moreover, the performance estimates should not vary broadly when the experiment is repeated. However, little is known about how accurate and reliable the performance estimates of MVTs tend to be.

Approach. We investigate the accuracy and the reliability of performance estimates when 10 MVTs (i.e., holdout validation, k-fold validation and bootstrap validation) are applied. We measure in terms of 5 threshold-dependent and -independent performance measures (e.g., precision, recall, AUC) and evaluate using different types of classification techniques.

Results. We find that (1) the advanced bootstrap validation is the most accurate and the most reliable model validation technique; and (2) the holdout family is the least accurate and most reliable model validation technique in terms of both threshold-dependent and threshold-independent performance measures.

5. PRACTICAL GUIDELINES

Our results indicates that (1) noise in defect datasets; (2) parameter settings of classification techniques; and (3) model validation techniques have a large impact on the predictions and insights of defect prediction models. Below, we offer practical guidelines for future defect prediction studies:

1. Researchers should experiment with a broader selection of datasets and metrics in order to maximize external validity [14].
2. Researchers should carefully mitigate collinearity issues prior to analysis in order to maximize internal and construct validity [14].
3. Researchers should carefully examine the choice of metrics when building defect prediction models in order not to produce under-performing models [14].
4. Researchers should clean mislabelled issue reports in order to improve the ability to identify defective modules [12].
5. Researchers should only interpret or make decisions based on the top influence metrics of defect prediction models when they are trained on noisy data [12].
6. Researchers should apply automated parameter optimization in order to improve the performance and reliability of defect prediction models [15].
7. Researchers should avoid using the holdout validation and instead opt to use the advanced bootstrap model validation technique in order to produce more accurate and reliable performance estimates [13].

6. CONCLUSIONS AND CONTRIBUTIONS

In this thesis, we investigate the impact that experimental components have on the predictions and insights of defect prediction models. Through case studies of systems that span both proprietary and open-source domains, we demonstrate that:

– The experimental components (e.g., metric family) that are used to construct defect prediction models share a stronger relationship with the reported performance than research group does.
– Noise generated by issue report mislabelling has an impact on the predictions and insights of defect prediction models.
– Parameter settings of classification techniques have an impact on the accuracy and reliability of the performance of defect prediction models.
– Model validation techniques have an impact on the accuracy and reliability of the performance estimates that are produced by defect prediction models.

Our findings lead us to conclude that experimental components of defect prediction modelling have a large impact on the predictions and insights that are derived from defect prediction models, suggesting that researchers should carefully select experimental components in order to produce more accurate and reliable defect prediction models.

7. PROGRESS & OUTLOOK

Our studies 1 and 2 have been published at the International Conference on Software Engineering (ICSE 2015, 2016), respectively. The preliminary study and the study 3 are under reviewed at the Transactions on Software Engineering (TSE). The author is conducting additional experiments for study 2 and planning to submit to a journal.

Acknowledgments. The author thanks Dr. Shane McIntosh, Dr. Ahmed E. Hassan, Dr. Kenichi Matsumoto, and anonymous reviewers for their constructive feedback.

8. REFERENCES

[1] C. Bird, A. Bachmann, E. Aune, J. Duffy, A. Bernstein, V. Filkov, and P. Devanbu. Fair and Balanced? Bias in Bug-Fix Datasets. In *Proceedings of the joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pages 121–130, 2009.

[2] B. Ghotra, S. McIntosh, and A. E. Hassan. Revisiting the impact of classification techniques on the performance of defect prediction models. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 789–800, 2015.

[3] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell. A Systematic Literature Review on Fault Prediction Performance in Software Engineering. *IEEE Transactions on Software Engineering*, 38(6):1276–1304, Nov. 2012.

[4] K. Herzig, S. Just, and A. Zeller. It's not a Bug, it's a Feature: How Misclassification Impacts Bug Prediction. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 392–401, 2013.

[5] S. Kim, H. Zhang, R. Wu, and L. Gong. Dealing with noise in defect prediction. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 481–490, 2011.

[6] M. Kuhn. caret: Classification and regression training. http://CRAN.R-project.org/package=caret, 2015.

[7] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch. Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings. *IEEE Transactions on Software Engineering*, 34(4):485–496, 2008.

[8] T. Menzies and M. Shepperd. Special issue on repeatable results in software engineering prediction. *Empirical Software Engineering*, 17(1-2):1–17, 2012.

[9] I. Myrtveit, E. Stensrud, and M. Shepperd. Reliability and validity in comparative studies of software prediction models. *IEEE Transactions on Software Engineering*, 31(5):380–391, 2005.

[10] F. Rahman and P. Devanbu. How, and why, process metrics are better. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 432–441, 2013.

[11] M. Shepperd, D. Bowes, and T. Hall. Researcher Bias: The Use of Machine Learning in Software Defect Prediction. *IEEE Transactions on Software Engineering*, 40(6):603–616, 2014.

[12] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, A. Ihara, and K. Matsumoto. The Impact of Mislabelling on the Performance and Interpretation of Defect Prediction Models. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 812–823, 2015.

[13] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto. An Empirical Comparison of Model Validation Techniques for Defect Prediction Model. *Under Review at IEEE Transactions on Software Engineering (TSE)*.

[14] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto. Comments on ”Researcher Bias: The Use of Machine Learning in Software Defect Prediction”. *PeerJ PrePrints*, 2015.

[15] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto. Automated Parameter Optimization of Classification Techniques for Defect Prediction Models. In *The International Conference on Software Engineering (ICSE)*, page To appear, 2016.