A Partial Replication Study: Does chronology matter in JIT defect prediction? (link)

Hadi Jahanshahi[1] hadi.jahanshahi@ryerson.ca

Dhanya Jothimani dhanya@ryerson.ca

Ayşe Başar ayse.bener@ryerson.ca

Mucahit Cevik mcevik@ryerson.ca

Original Paper:

Are Fix-Inducing Changes a Moving Target? A Longitudinal Case Study of Just-In-Time Defect Prediction (link)

Shane McIntosh and Yasutaka Kamei

WHAT: In this work, we aim to investigate the effect of code change properties on JIT models over time. We also study the impact of using recent data as well as all available data on the performance of JIT models. Further, we analyze the effect of weighted sampling on the performance of fix-inducing properties of JIT models. For this purpose, we used datasets from four open-source projects, namely Eclipse JDT, Mozilla, Eclipse Platform, and PostgreSQL.

WHY: Despite several advantages of JIT prediction models, they require a large amount of historical data for improved model performance. Also, it works on the assumption that the properties of future events are similar to the properties of previous ones. However, this assumption may not hold true due to the dynamic nature of software development projects. Hence, in this study, first, we investigated whether the properties of fix-inducing changes remain consistent with the evolution of the system over time, and second, we analyzed the importance of such a potential evolution in JIT defect prediction domain.

HOW: McIntosh and Kamei examined whether the properties of past events such as fix-inducing changes are similar to the properties of the future ones. To achieve this overall objective, the researchers formulated three research questions. To address the research questions, six code properties, namely, size, diffusion, history, author and reviewer experiences, and review, were used for training JIT models on two open-source systems, i.e. Qt and OpenStack. These systems exhibited three characteristics such as traceability, rapidly evolving and code review policy.

WHERE: The original study was conducted on two open-source systems. In order to generalise/validate results of the original study and to better understand the evolving nature of fix-inducing changes, we focus on historical datasets of other software systems, namely Eclipse JDT, Mozilla, Eclipse Platform, and PostgreSQL. In our study, we defined one additional research question, which investigates different hypotheses to check the possibility of improvement in the performance of JIT models while considering the recency of the data. Furthermore, we used Random Forest instead of Logistic Regression to investigate the sensitivity of the result to the selected model.

DISCUSSION: Replication of the original study enabled us to confirm the impact of chronology on the importance of each family of code change properties. Furthermore, the outcome of RQ1 from our study was different than that of the original paper, indicating the need for further research to make practitioners confident about the generalizability of the issue. Finally, we proposed a new weighted sampling approach to complement the deduction made by the original study. The proposed approach is capable of augmenting the calibration ability of JIT models. For future research, this work can be replicated using systems that are developed in other contexts or the methods that are not covered by either of the works.

[1] *corresponding author, GitHub ID: HadiJahanshahi;*
GitHub page of the replication study: https://github.com/HadiJahanshahi/JITChronology