Defectors: A Large, Diverse Python Dataset for Defect Prediction

Parvez Mahbub
Dalhousie University
parvezmrobin@dal.ca

Ohiduzzaman Shuvo
Dalhousie University
ohiduzzamanshuvo@dal.ca

Mohammad Masudur Rahman
Dalhousie University
masud.rahman@dal.ca

***Abstract*—Defect prediction has been a popular research topic where machine learning (ML) and deep learning (DL) have found numerous applications. However, these ML/DL-based defect prediction models are often limited by the quality and size of their datasets. In this paper, we present Defectors, a large dataset for just-in-time and line-level defect prediction. Defectors consists of ≈ 213K source code files (≈ 93K defective and ≈ 120K defect-free) that span across 24 popular Python projects. These projects come from 18 different domains, including machine learning, automation, and internet-of-things. Such a scale and diversity make Defectors a suitable dataset for training ML/DL models, especially transformer models that require large and diverse datasets. We also foresee several application areas of our dataset including defect prediction and defect explanation.**

***Index Terms*—Defect Prediction, Just-in-Time, Dataset, Software Engineering**

I. Introduction

A software defect is an incorrect step, process, or data definition in a computer program that prevents the program from working correctly [1]. Software defects are informally called software bugs. They cost the global economy billions of dollars every year [2], [3]. Despite the adoption of various software quality assurance (SQA) practices, defects may still sneak into official releases [4], [5]. Furthermore, a recent work [6] shows that only ≈ 3% code lines of the whole release could lead to many of the bugs. Therefore, prioritizing SQA efforts for highly risky areas of source code is essential to ensure the high quality of a software release.

Defect prediction has been a popular research topic for the last few decades. It identifies the defects in software code before releasing the software to end users. It can also help prioritize the SQA efforts. Defects can be predicted at different abstraction levels such as module [7], [8], file [9], [10], method [11], and line [6], [12]–[14]. In recent years, just-in-time (JIT) defect prediction [14]–[20] also has gained significant attention, which predicts the defects just at the time of committing software changes. Thus, a combination of line-level defect prediction and JIT defect prediction can provide a fine-grained location of a software defect.

Over the past few years, deep-learning models have been used for both line-level defect prediction [6] and JIT defect prediction [18], [19]. Deep learning models provide state-of-the-art performance in various tasks of software engineering including bug localization [21], [22] and bug explanation [23]. However, their performance in JIT defect prediction is sub-optimal. Deep-learning-based tools such as DeepJIT [18] and CC2Vec [19] cannot outperform simpler models such as logistic regression. These models can be limited by the size and quality of their datasets. First, the performance of ML/DL models often scales with the size of their dataset [24], [25]. However, most of the existing datasets used in defect prediction might not be large enough [26]. Second, these datasets also suffer from the class imbalance problem containing only 5%-26% defective instances [15], [20], [26]. Such an imbalance could lead to sub-optimal performance with any deep-learning models. Third, these datasets were constructed either from a small number of projects [15] or the projects from a single organization [26], [27]. Such a choice limits the capability of these models to generalize their performances across different domains and organizations.

To mitigate the challenges with existing datasets, in this paper, we present *Defectors* – a large-scale dataset, containing both source code and their changes from 24 popular Python projects across 18 domains and 24 organizations. We carefully identify defective source code files and their code changes, following five levels of noise filtration recommended in the literature. Our dataset contains ≈ 213K source code files (≈ 93K defective and ≈ 120K defect-free). It is suitable for training large models on the task of defect prediction that has the potential to provide high performance.

Defectors stands out from similar datasets in the following aspects.

1) **Size:** To the best of our knowledge, Defectors is the largest defect prediction dataset and is twice in size as the previous largest dataset (i.e., ≈ 106K) [26].
2) **Class Balance:** It maintains a near 1:1 ratio between defective and defect-free instances in the training set, where existing datasets contain only 5%-26% defective instances [15], [20], [26].
3) **Diversity in Application:** Defectors uses 24 projects from 18 application domains and 24 organizations, where existing datasets either use a small number of projects (< 10) [13], [15] or the projects from only one organization (e.g., Apache) [26], [27].
4) **Diversity in Platform:** Our dataset is based on Python projects, whereas nearly all existing datasets were constructed from Java-based projects. Thus, it diversifies the existing collection of defect prediction datasets.

The dataset is publicly available at the following link: https://doi.org/10.5281/zenodo.7708984