

Software Defect Prediction and Bugs Classification

K.T. Liyanage^{1*}, N. H. Amarasinghe¹, S. Nuzrath¹, D. P. Madanayake¹, K. Suriyawansa² and N. Kodagoda²

^{1*}*Department of Information Technology, Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka*

²*Department of Software Engineering, Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka*

Defect prediction empathizes a main role in the Software Development Life Cycle. Having defects in source code is unavoidable but identifying and reducing the bugs in source code in early stages save time and effort. The quality of the source code is primarily measured based on the defect rate. Software defects have a strong relationship with software metrics. As a developer once they build the code, it redirects to the quality engineers to test the code, and if there is any fault with the software, they will send it back to the developers with their comments. Since the software was built by many developers, each developer has to go through the code to find the faultiness. This process will have a significant efficiency improvement if the developer responsible for the bug is identified at the beginning. This tool provides a practical solution, where once we get the feedback from the quality assurance engineers, tool labels the developers(n), and redirect all the comments to the relevant bucket. Then the developers can easily identify their faults and fix bugs. To understand the status of the code, researchers collected several metrics which are related to defect prediction such as Cyclomatic Complexity values and Halstead Complexity values, then using the Principle Component Analysis, it identifies the most relevant software metrics for defect prediction and builds the dataset. Using the dataset researchers developed a machine learning model to predict the code status, Whether the code is in Good, Moderate, or Weak Level. Natural Language Processing used to analyse the Git issues, with the aid of the Latent Dirichlet allocation algorithm, it is based on clustering and create needed categories for a given input(n). Once the user gives the link of the source code, the tool identifies the defect rate, responsible developer for each bug, most committed authors, and the frequencies of most used words. The result shows that the tool solves the practical problem more accurately in the programming environment.

Keywords: Defect Prediction, Software Metrics, Git Issues, Defect rate, Machine