

ANALYSIS OF SOFTWARE DEFECT PREDICTION USING OPTIMIZED DUAL SCALED SELF-SPARSE GUIDED OCTAVE CONVOLUTION NETWORK

Suresh Yallamati [1*], Shaheda Akthar[2]

[1*]Research Scholar, Department of Computer Science and Engineering, Acharya Nagarjuna University, Andhra Pradesh, India

[2]Associate Professor, Department of Computer Science, Government College for Women (A), Guntur, Andhra Pradesh, India

Corresponding Author: Suresh Yallamati
Email: sureshyallamati734@gmail.com
Doi: 10.34107/TGHN9103**020014**

ABSTRACT

This research is driven by the necessity to have more precise defect prediction techniques that would minimize false alarms, leading to improved software development procedures. The analysis of software defect prediction with the use of optimized dual-scaled self-sparse guided octave convolution network is the proposed method to utilize attention mechanisms and give the first priority to the relevant features that will enhance the capability of the model to distinguish between non-defective and defective code changes. To provide a complete analysis, this research used defect data sources of MIS and KC2. Min-max scaling and data log transformation are some of the pre-processing methods that were used to improve data quality. The Hybrid Kookaburra Alpine Skiing Optimization Algorithm was used to select the features, which were relevant in defect prediction. Afterwards, the Dual Scaled Self-Sparse Guided Octave Convolution Network model was created that combines attention mechanisms, multi-scale features to predict any defects accurately and improve the quality assurance of the software. Moreover, the Mud Ring Algorithm was also implemented to adjust the network, and the accuracy of prediction was even better. These works represent the improvements of deep learning-based defect prediction procedures, which hold a potential of further improvement of the software quality evaluation and guarantee. The suggested approach achieves a greater accuracy of 99.64%.

Keywords: Software Defect prediction, Dual Scaled Self-Sparse Guided Octave Convolution Network, mud ring optimization, Hybrid Kookaburra Alpine Skiing Optimization.

INTRODUCTION

In the world of software engineering, one of the most intriguing research areas has been software fault prediction [1-2]. Semantic flaws raise application expenses, cause problems for users, and might have major ramifications during deployment, according to consistent reports. Thus, it is imperative that bugs be localized and fixed early in the software development process. Different methods have been put forth to build models that can determine whether or not a source code includes errors [3-4]. These studies may be broadly classified into two directions: the first uses machine learning techniques on manually constructed software metrics data to extract features from source code, while the other uses deep learning and program tree representations to automatically learn defect attributes [5-6].

The conventional techniques concentrate on generating and assembling software features. The mainstream of product metrics relies on source code statistics. Halstead metrics are derived from the number of operators and operands; CK metrics are determined by function and inheritance counts [7–8]. Nonetheless, a number of studies and polls indicate that the current metrics frequently fall short of accurately capturing the program semantics. Consequently, the classifier performance is not that great even after a lot of work, such as adopting robust learning techniques and improving the data [9-10].