International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER SSRN

IJIEMR Transactions, online available on 19th Nov 2022. Link :http://www.ijiemr.org/downloads.php?vol=Volume-11&issue=Issue 11

DOI: 10.48047/IJIEMR/V11/ISSUE 11/06

Title DEFECT PREDICTION IN SOFTWARE USING SPIDERHUNT OPTIMIZATION TECHNIQUE

Volume 11, ISSUE 11, Pages: 38-42

Paper Authors

M.Prashanthi,Dr.M.Chandra Mohan

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DEFECT PREDICTION IN SOFTWARE USING SPIDERHUNT OPTIMIZATION TECHNIQUE

M.Prashanthi, Research Scholar at JNTUH,Asst Professor,CMR Engineering College

Dr.M.Chandra Mohan,Professor Of Cse,JNTUH,Kukatpally

Abstract

Software in the computer is significant in performing the important tasks in computer hardware and also plays a vital role in improving the efficiency of the daily assets of human individuals. The quality of the software is threatened due to various defects and it must be determined and resolved for the effective functioning of the system. In the Software development life cycle, the software defect prediction assists the system to enhance their quality and various researches are undertaken for the effective prediction of defects in software. In this research, the defects in the software are predicted using the deep CNN classifier by effectively optimizing the classifier using spiderhunt optimization. The effective communication and hunting characteristics of the spiderhunt are employed for tuning the classifier that boosts the classifier performance. The proposed spiderhunt optimization not only optimizes the classifier but also plays a significant role in the feature selection for the extraction of necessary features that helps in the defect prediction. The proposed spiderhunt optimization achieved the improvement in terms of accuracy, precision, recall and f-measure and is proved to be quite efficient compared to state of art methods.

Keywords: Deep CNN classifier, spiderhunt optimization, software defect prediction, threat, communication.

1. Introduction

In the recent era, software defect prediction in software reliability engineering gained considerable attentiveness from both industrial and academic individuals [11]-[14][4]. The advancement in the development of software systems proportionally increases the complexities and the defect that occurs in the software due to the facts like lack of understanding of the software requirements, unproductive development, and improper experience in developing software [1]. The defects present in the system provide inefficient results in the software after the execution of the products, which in turn causes a significant impact on the lifestyle routine of individuals. The development of software prediction models acts as an auspicious solution for assuring the quality of the software as well as the presence of defectiveness in the system model. Similarly, the software prediction models help in reducing expenses and improving the reliability of the software. Software quality assurance teams availing the software defect prediction model can detect the defects at the earliest time without error because manual detection can cause error even though complete consciousness is applied [7]. Researchers provided high-quality software systems by manually defining the features that differentiate the defected and non-defected files [3][4]. Classification of data is a

crucial function in the prediction of defects present in the software model that involves the categorization of software metrics, fault-prone, code attributes, non-fault prone, or any other attributes necessary for the prediction, and the information from the previously developed models are also utilized for this classification [15][10].

2.1 Literature review

The various works relying on finding the defects in software are enumerated as follows: Shiqi Tang *et al.*[1] designed a transfer learning algorithm TSboostDF that provided statistically significant experimental results. Although it worked well, it is not suitable for all the datasets. Hao Wang *et al.*[2] flourished a defect detecting model named gated hierarchical long short-term memory networks (GH-LSTMs) that had the capability to collect the semantic features as well as traditional features but the defect is predicted using fewer number of instances. Song Wang *et al.*[3] initiated a representation-learning algorithm that measured the metrics based on two file level, which enhances the performance but the prediction of bugs with more accuracy cause difficulties. Tanujit Chakraborty *et al.*[4] prospered a Hellinger net model that dwelled with the data imbalance with adequate performance but the model doesn't deal with unlabelled data. Hongliang Liang *et al.*[5] introduced Seml for the defect prediction that performed software traceability as well as detected vulnerabilities but the method can be enhanced by collecting more semantic information. Liu Yang *et al.*[6] detected the defects by the hybridization of the particle swarm and sparrow search optimization that provided a good convergence rate and stability yet it plays a minor role in the defect prediction. Chi Yu *et al.*[7] originated homomorphic encryption to software defect prediction model (HOPE) that not only detected the defects in the system but also protected the privacy of the users but the method has the capability to encrypt only the integers. Lucija Sikic *et al.*[8] innovated aggregated change metrics, that detected the potential defects with notable metrics value but furthermore there is a need for improvement in this method to achieve higher and better performances

4. Proposed spiderhunt optimization-based deep CNN for the software defect prediction

The quality, reliability, and cost solely depend on the software that works more efficiently, but the occurrence of defects influences the overall outcome of the device. Hence an effective software defect prediction module is developed using the deep CNN network in this research. Initially, the data from the PROMISE dataset is collected and the important features from the data are extracted using the proposed spiderhunt optimization. In feature selection, the spiderhunt optimization helps in picking up the significant features, and furthermore, the selected data is subjected to the optimized deep CNN classifier. The optimization applied to the deep CNN helps in enhancing the weights and bias by optimally tuning the parameters reliable for attaining desired output. The enhancement made in the defect prediction using deep CNN is evaluated using the metrics accuracy, precision, recall, and f-measure, which is more efficient. The

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

representation of the software defect prediction model is shown in figure 1.

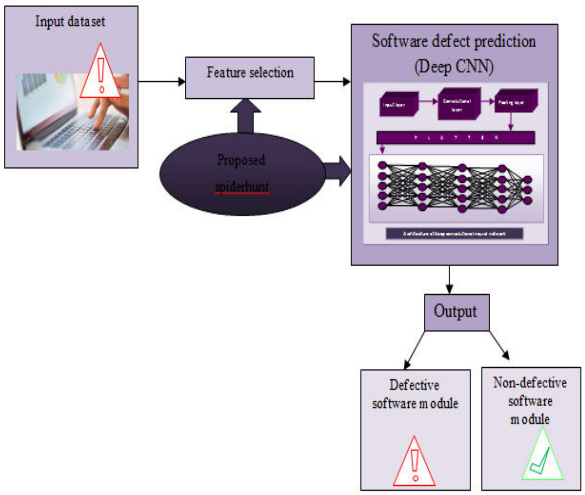

Figure 1: System model for the software defect prediction using optimized deep CNN

Table 1: Algorithmic procedures for the proposed spiderhunt optimization

S.NO	Pseudocode for the spiderhunt optimization
1	Initialization
2	Social grading
3	Locking phase
4	Snaring phase
5	Offensive phase
6	Selective phase
7	Communicative phase
8	Initialization
9	Initialize: λ
10	Social grading
11	Order based on best solution: λ_1, λ_2, λ_3
12	Locking phase
13	Bound the prey
14	$\vec{Y} = \lvert \vec{R} \cdot \vec{Q}_x(n) - \vec{Q}(n) \rvert$
15	Determine coefficient vectors: $\vec{H}$, $\vec{R}$
16	Snaring phase
17	Update the position of wolves
18	$\vec{Q}(n+1) = \frac{\vec{Q}_1 + \vec{Q}_2 + \vec{Q}_3}{3}$

19	Offensive phase
20	if $\left(\lvert\vec{H}\rvert<1\right)$
21	Starts attack
22	else
23	Search
24	Selective phase
25	if $\left(\vec{R}<1\right)$
26	Diverge
27	else
28	Unite
29	Communicative phase
30	Evaluate the size, distance
31	$D_e = W_f E^{g_{ef}^2}$
32	Enhancing phase
33	$C_{best} = W_f E^{g_{ef}^2}\left(\vec{Q}(n+1)\right)$

6. Conclusion

Software defect prediction using the proposed spiderhunt optimization based deep CNN is performed in this research. The effective defect identification in the software helps in improving the quality of the product along with that the cost of the system is also reduced if the defects are identified earlier. Extensive testing is performed in the testing phase of the software to detect the defects and the usage of proposed spiderhunt optimization based deep CNN reduced the time consumption along with that the prone to error are also reduced. The proposed classifier helps in both feature extraction mechanism and tuning the classifier that reduces the complexity of the network. The proposed spiderhunt optimization achieved the improvement rate of 1.009 %, 1.083 %, 0.578 %, and 1.01 % in terms of accuracy, precision, recall and f-measure. In this research the presence of defects are detected and in future the defects could be identified and eliminated using the classifiers, which will be more proficient.

References:

[1] S. Tang, S. Huang, C. Zheng, E. Liu, C. Zong and Y. Ding, "A novel cross-project software defect prediction algorithm based on transfer learning," in Tsinghua Science and Technology, Vol. 27, no. 1, pp. 41-57, 2022.

[2] H. Wang, W. Zhuang and X. Zhang, "Software Defect Prediction Based on Gated Hierarchical LSTMs," in IEEE Transactions on Reliability, vol.70, no.2, pp.711-727, 2021.

[3] S. Wang, T. Liu, J. Nam and L. Tan, "Deep Semantic Feature Learning for Software Defect Prediction," in IEEE Transactions on Software Engineering, Vol. 46, no.12, pp.1267-1293, 2020.

[4] T. Chakraborty and A. K. Chakraborty, "Hellinger Net: A Hybrid Imbalance Learning Model to Improve Software Defect Prediction," in IEEE Transactions on Reliability, Vol.70, no.2, pp. 481-494, 2021.

[5] H. Liang, Y. Yu, L. Jiang and Z. Xie, "Seml: A Semantic LSTM Model for Software Defect Prediction," in IEEE Access, vol.7, pp.83812-83824, 2019.

[6] L. Yang, Z. Li, D. Wang, H. Miao and Z. Wang, "Software Defects Prediction Based on Hybrid Particle Swarm Optimization and Sparrow Search Algorithm," in IEEE Access, vol.9, pp. 60865-60879, 2021.

[7] C. Yu, Z. Ding and X. Chen, "HOPE: Software Defect Prediction Model Construction Method via Homomorphic Encryption," in IEEE Access, vol. 9, pp. 69405-69417, 2021.

[8] L. Šikić, P. Afrić, A. S. Kurdija and M. ŠIlić, "Improving Software Defect Prediction by Aggregated Change Metrics," in IEEE Access, vol. 9, pp. 19391-19411, 2021.

[9] J. Li, P. He, J. Zhu and M. R. Lyu, "Software Defect Prediction via Convolutional Neural Network," in 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), pp. 318-328, 2017.

[10] Lessmann, Stefan, Bart Baesens, Christophe Mues, and Swantje Pietsch, "Benchmarking classification models for software defect prediction: A proposed framework and novel findings," IEEE Transactions on Software Engineering, Vol.34, no.4, pp.485-496, 2008.

[11] Siers, Michael J., and Md Zahidul Islam, "Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem," in Information Systems, Vol. 51, pp.62-71, 2015.

[13] Jing, Xiao-Yuan, Fei Wu, Xiwei Dong, and Baowen Xu, "An improved SDA based defect prediction framework for both within-project and cross-project class-imbalance problems," in IEEE Transactions on Software Engineering, Vol.43, no.4, pp.321-339, 2016.

[14] Fenton, Norman E., and Martin Neil, "A critique of software defect prediction models," in IEEE Transactions on software engineering, Vol.25, no.5, pp.675-689, 1999.

[15] Schneidewind, Norman F, "Methodology for validating software metrics," in IEEE Transactions on software engineering, Vol.18, no.5, pp.410-422, 1992.