International Journal of Scientific Modern Research and Technology
(Volume: 11, Issue: 1, Number: 5)
ISSN: 2582-8150

Software Defect Prediction using Deep Learning: A Perspective View

Saloni Yadav[1], Dr. Manoj Lipton[2], Chetan Agrawal[3]
[1]Research Scholar, [2]Assistant Professor, [3]Head and Professor
Department of CSE, RITS, Bhopal, India

Abstract: Defect prediction is one of the key challenges in software development and programming language research for improving software quality and reliability. The problem in this area is to properly identify the defective source code with high accuracy. Developing a fault prediction model is a challenging problem, and many approaches have been proposed throughout history. The recent breakthrough in machine learning technologies, especially the development of deep learning techniques, has led to many problems being solved by these methods. Our survey focuses on the deep learning techniques for defect prediction. We analyze the recent works on the topic, study the methods for automatic learning of the semantic and structural features from the code, discuss the open problems and present the recent trends in the field.

Keywords: defect prediction; anomaly detection; program analysis; code understanding; neural networks; deep learning

1. INTRODUCTION

According to the IEEE Standard Classification for Software Anomalies [1], a software defect is "an imperfection or deficiency in a work product where that work product does not meet its requirements or specifications and needs to be either repaired or replaced". Software defects can cause different problems. Common ways to find software defects are manual testing and code review. The main drawback of these methods is that they are quite expensive in terms of time and effort.

The automatic approaches to the Software Defect Prediction (SDP) would allow one to reduce the costs and improve quality of the software projects. Thus, Software Defect Prediction is an important problem in the fields of the software engineering and programming language research. The task is to identify the defective code with high accuracy (in terms of the precision and recall) [2]. The development and breakthrough of machine learning led to the fact that many tasks can be solved by the these methods. Recent advances in the fields of artificial neural networks and machine learning, as well as the increasing power of the modern computers (such as supercomputers based on GPUs with AI accelerating modules), allowed new concepts, such as deep learning, to emerge [3].

The main idea is that an artificial neural network with multiple layers is capable of progressively extracting the higher-level features from the original data to solve complex problems [4]. For the problem of software defect prediction, the researchers have proposed the representation-learning algorithms to learn semantic representations of programs automatically and use this representation to identify the defect-prone code. Using these implicit features shows better results than the previous approaches based on the explicit features, such as the code metrics [5].

II. LITERATURE WORK

Automated software defect prediction (SDP) methods are increasingly applied, often with the use of machine learning (ML) techniques. Yet, the existing ML-based approaches require manually extracted features, which are cumbersome, time-consuming and hardly capture the semantic information reported in bug reporting tools. Deep learning (DL) techniques provide practitioners with the opportunity to automatically extract and learn from more complex and high-dimensional data. (Görkem Giray, Kwabena Ebo Bennin, Ömer Köksal, Önder Babur, Bedir Tekinerdogan; 2023)

Defect prediction is one of the key challenges in software development and programming language research for improving software quality and

reliability. The problem in this area is to properly identify the defective source code with high accuracy. Developing a fault prediction model is a challenging problem, and many approaches have been proposed throughout history. (Akimova, E.N., Bersenev, A.Y., Deikov, A.A., Kobylkin, K.S., Konygin, A.V., Mezentsev, I.P., Misilov; 2021)

Recent advances in machine learning have stimulated widespread interest within the Information Technology sector on integrating AI capabilities into software and services. This goal has forced organizations to evolve their development processes. We report on a study that we conducted on observing software teams at Microsoft as they develop AI-based applications. (Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan; 2019)

In neural networks literature, there is a strong interest in identifying and defining activation functions which can improve neural network performance. In recent years there has been a renovated interest of the scientific community in investigating activation functions which can be trained during the learning process, usually referred to as "trainable", "learnable" or "adaptable" activation functions. They appear to lead to better network performance. (Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, Roberto Prevete; 2021)

The software development life cycle generally includes analysis, design, implementation, test and release phases. The testing phase should be operated effectively in order to release bug-free software to end users. In the last two decades, academicians have taken an increasing interest in the software defect prediction problem, several machine learning techniques have been applied for more robust prediction. (Ömer Faruk Arar, Kürşat Ayan; 2015)

In recent years, data science has been used extensively to solve several problems and its application has been extended to several domains. This paper summarises the literature on the synergistic use of Software Engineering and Data Science techniques (e.g. descriptive statistics, inferential statistics, machine learning, and deep learning models) for predicting defects in software. It shows that there is a variation in the use of data science techniques and limited reasoning behind the choice of certain machine learning models but also, in the evaluation of the obtained results. (Farah Atif, Manuel Rodriguez, Luiz J. P. Araújo, Utih Amartiwi, Barakat J. Akinsanya & Manuel Mazzara; 2021)

Systematic literature studies are commonly used in software engineering. There are two main ways of conducting the searches for these types of studies; they are snowballing and database searches. In snowballing, the reference list (backward snowballing - BSB) and citations (forward snowballing - FSB) of relevant papers are reviewed to identify new papers whereas in a database search, different databases are searched using predefined search strings to identify new papers. Objective: Snowballing has not been in use as extensively as database search. Hence it is important to evaluate its efficiency and reliability when being used as a search strategy in literature studies. (Deepika Badampudi, Claes Wohlin, Kai Petersen; 2015)

Software fault/defect prediction assists software developers to identify faulty constructs, such as modules or classes, early in the software development life cycle. There are data mining, machine learning, and deep learning techniques used for software fault prediction. We perform analysis of previously published reviews, surveys, and related studies to distill a list of questions. These questions were either answered in the past but needed a fresh look or they were not considered at all. We justify why answers to newly added questions are important and divide previous work based on data mining, machine learning, and deep learning and compare their performance. (Iqra Batool, Tamim Ahmed Khan; 2022)

Context: Recent studies have shown that performance of defect prediction models can be affected when data sampling approaches are applied to imbalanced training data for building defect prediction models. However, the magnitude (degree

 ISSN: 2582-8150

and power) of the effect of these sampling methods on the classification and prioritization performances of defect prediction models is still unknown. Goal: To investigate the statistical and practical significance of using resampled data for constructing defect prediction models. Method: We examine the practical effects of six data sampling methods on performances of five defect prediction models. (Kwabena Ebo Bennin; Jacky Keung; Akito Monden; Passakorn Phannachitta; Solomon Mensah; 2017)

III. PROBLEM IDENTIFICATION

Following are the problem identification on the basis of existing work [1]:

- The identification of relevant software bugs is not perfectly retrieved.
- The retrieval of a software bug is not perfectly identified.
- The unidentified software bug may detect due to low accuracy.

IV. RESEARCH OBJECTIVES

Following are the objectives of the proposed work:

- To improve precision for perfect retrieval of relevant software bugs.
- To improve recall for perfectly relevant software bugs in the retrieval process.
- To improve accuracy for exactness of software bug detection.

IV. COMPARATIVE ANALYSIS

Table 1: Analysis of Different Techniques

Techniques	Data set used	Advantages	Limitations
Artificial Neural network [3]	NASA AR1,AR6 And MDP	No need to know metrics relationships. It has self-learning capability therefore get more accuracy	It cannot manage imprecise information
Support Vector Machine [4]	NASAAR1 , AR6	Using different kernel function it gives better prediction result	Not suitable for large number of software metrics
Decision Tree [5]	NASA AR1,AR6	Performing operation on tree structure therefore more accurate result compare to others	Construction of decision tree is complex
Association Rule [6]	NASA MDP repository	Generated rules using historical data and predict defect	Require Continues value of software metrics
Clustering [10]	NASA MDP repository	It suitable for small dataset	Dataset should be unlabeled

 ISSN: 2582-8150

V. PROPOSED PREDICTION MODEL

The outline of the proposed prediction model SVM-MMN (Support Vector Machine with Min-Max Normalization) is as follows:

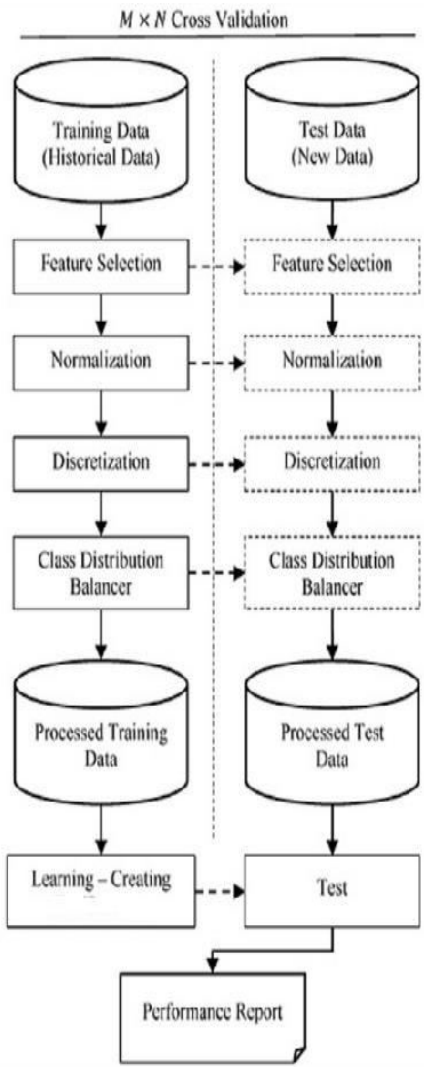

Figure 1: Proposed Model of SVM-MMN (Proposed Methodology)

VI. CONCLUSIONS

One of the major challenges in modern software engineering is predicting defective code. Recent developments in the field of machine learning, especially the multi-layered neural networks and deep learning algorithms, provide powerful techniques, which utilize learning algorithms for representations of the source code that captures semantic and structural information.

This survey presents the latest research progress in software defect prediction using the deep learning techniques, such as the Transformer architectures. We formulate the main difficulties of the defect prediction problem as lack of data and complexity of context and discuss the ways to alleviate these problems.

REFERENCES

[1] Görkem Giray, Kwabena Ebo Bennin, Ömer Köksal, Önder Babur, Bedir Tekinerdogan, “On the use of deep learning in software defect prediction”, The Journal of Systems & Software, 2023.
[2] G. P. Bhandari and R. Gupta, “Machine learning based software fault prediction utilizing source code metrics,” in 250 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 2018, pp. 40–45.
[3] R. Malhotra, “A systematic review of machine learning techniques for software fault prediction,” Appl. Soft Comput., vol. 27, pp. 504–518, Feb. 2015.
[4] L. Son et al., “Empirical Study of Software Defect Prediction: A Systematic Mapping,” Symmetry (Basel)., vol. 11, no. 2, p. 212, Feb. 2019.
[5] C. W. Yohannese and T. Li, “A Combined-Learning Based Framework for Improved Software Fault Prediction,” Int. J. Comput. Intell. Syst., vol. 10, no. 1, p. 647, Dec. 2017.
[6] A. Hudaib et al., “ADTEM-Architecture Design Testability Evaluation Model to Assess Software Architecture Based on Testability Metrics,” J. Softw. Eng. Appl., vol. 08, no. 04, pp. 201–210, Apr. 2015.
[7] S. Elmidaoui, L. Cheikhi, and A. Idri, “Towards a Taxonomy of Software Maintainability Predictors,” Springer, Cham, 2019, pp. 823–832.
[8] M. Riaz, E. Mendes, and E. Tempero, “A systematic review of software maintainability prediction and metrics,” in 2009 3rd International Symposium on Empirical Software Engineering and Measurement, 2009, pp. 367–377.

[9] "PROMISE DATASETS PAGE." [Online]. Available: http://promise.site.uottawa.ca/SERepository/datasetspage.html. [Accessed: 01-Jul-2019].
[10] R. Malhotra and S. Kamal, "An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data," Neurocomputing, vol. 343, pp. 120–140, May 2019.
[11] 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology. .
[12] P. Oman and J. Hagemeister, "Construction and testing of polynomials predicting software maintainability," J. Syst. Softw., vol. 24, no. 3, pp. 251–266, Mar. 1994.
[13] T. Anderson, P. A. Barrett, D. N. Halliwell, and M. R. Moulding, "Software Fault Tolerance: An Evaluation," IEEE Trans. Softw. Eng., vol. SE-11, no. 12, pp. 1502–1510, Dec. 1985.
[14] I. U. Nisa and S. N. Ahsan, "Fault prediction model for software using soft computing techniques," in 2015 International Conference on Open Source Systems & Technologies (ICOSST), 2015, pp. 78–83.
[15] S. N. Ahsan and F. Wotawa, "Fault Prediction Capability of Program File's Logical-Coupling Metrics," in 2011 Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement, 2011, pp. 257–262.
[16] D. Radjenović, M. Heričko, R. Torkar, and A. Živkovič, "Software fault prediction metrics: A systematic literature review," Inf. Softw. Technol., vol. 55, no. 8, pp. 1397–1418, Aug. 2013.
[17] R. Malhotra, "Comparative analysis of statistical and machine learning methods for predicting faulty modules," Appl. Soft Comput., vol. 21, pp. 286–297, Aug. 2014.
[18] Z. Xu et al., "Software defect prediction based on kernel PCA and weighted extreme learning machine," Inf. Softw. Technol., vol. 106, pp. 182–200, Feb. 2019.
[19] J. Moeyersoms, E. Junqué de Fortuny, K. Dejaeger, B. Baesens, and D. Martens, "Comprehensible software fault and effort prediction: A data mining approach," J. Syst. Softw., vol. 100, pp. 80–90, Feb. 2015.
[20] Q. Yu, J. Qian, S. Jiang, Z. Wu, and G. Zhang, "An Empirical Study on the Effectiveness of Feature Selection for Cross-Project Defect Prediction," IEEE Access, vol. 7, pp. 35710–35718, 2019.