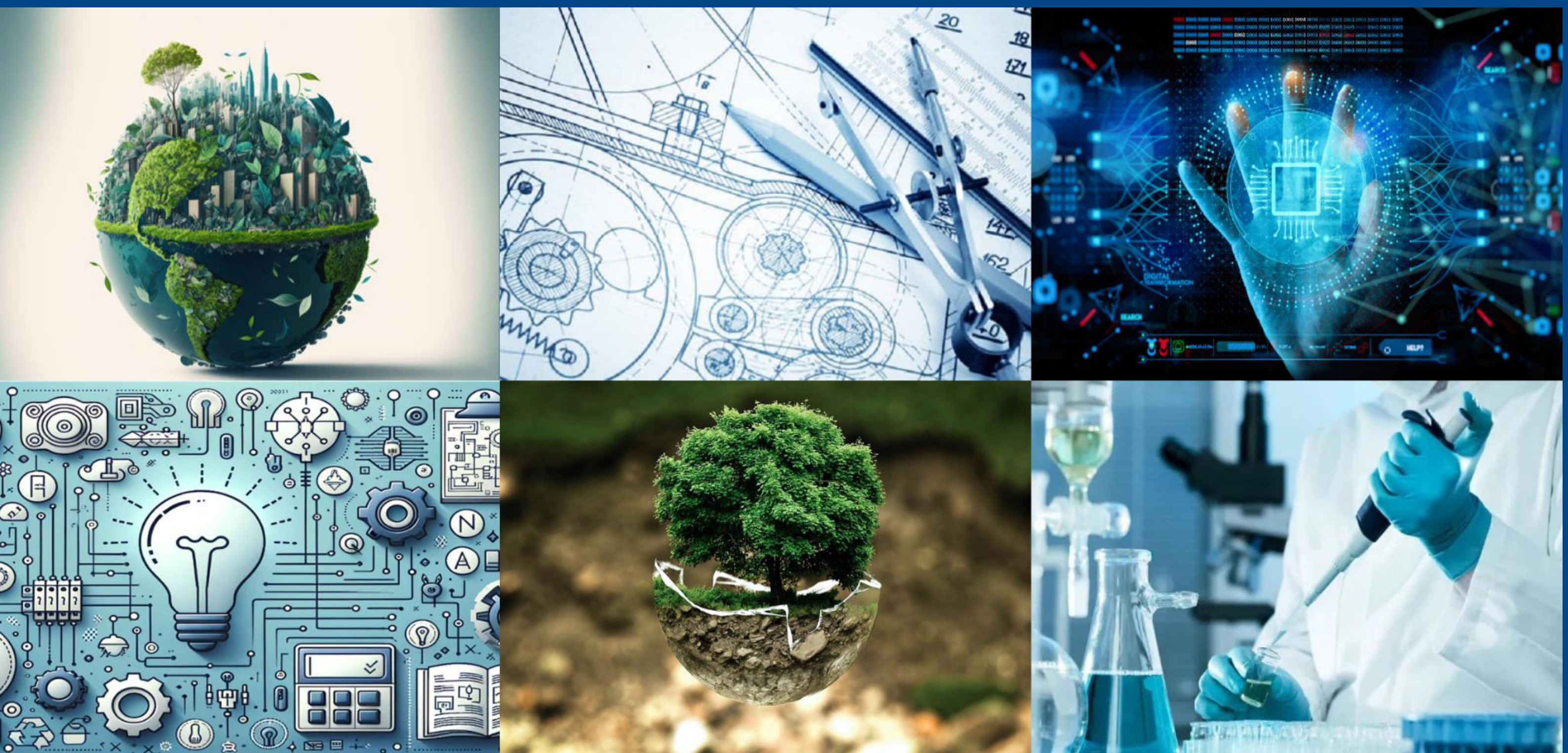
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 3, March 2025

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Predicting Bug Reports using Nature Based and Ensemble Machine Learning Model

Dr. D.J. Samatha Naidu, K. Venkata Ramya, T. Venkata Sai Himasri

Professor, Dept. of MCA, APGCCS, New Boyanapalli, Rajampet, A.P, India

Assistant Professor, Dept. of MCA, APGCCS, New Boyanapalli, Rajampet, A.P, India

PG Student, Dept. of MCA, APGCCS, New Boyanapalli, Rajampet, A.P, India

ABSTRACT: In software development systems, the maintenance process of software systems attracted the attention of researchers due to its importance in fixing the defects discovered in the software testing by using bug reports (BRs) which include detailed information like description, status, reporter, assignee, priority, and severity of the bug and other information. The main problem in this process is how to analyze these BRs to discover all defects in the system, which is a tedious and time-consuming task if done manually because the number of BRs increases dramatically. Thus, the automated solution is the best. Most of the current research focuses on automating this process from different aspects, such as detecting the severity or priority of the bug. However, they did not consider the nature of the bug, which is a multi-class classification problem. This project solves this problem by proposing a new prediction model to analyze BRs and predict the nature of the bug. The proposed model constructs an ensemble machine learning algorithm using natural language processing (NLP) and machine learning techniques. We simulate the proposed model by using a publicly available dataset for two online software bug repositories (Mozilla and Eclipse), which includes six classes: Program Anomaly, GUI, Network or Security, Configuration, Performance, and Test-Code.

I. INTRODUCTION

In software development, defects in response to client requirements can lead to system failures, often discovered during software testing. These defects, commonly referred to as software faults, arise due to errors in the development process and can compromise software quality. Predicting software defects is a crucial aspect of software engineering, aiming to identify faulty modules early in the development cycle. Effective defect prediction models play a significant role in reducing development costs, improving maintenance efficiency, and enhancing software reliability. By detecting defects early, organizations can prioritize testing efforts, optimize resource allocation, and ensure the delivery of high-quality software. Traditional defect prediction methods often rely on static analysis and historical defect data, but they face challenges in handling complex software architectures.

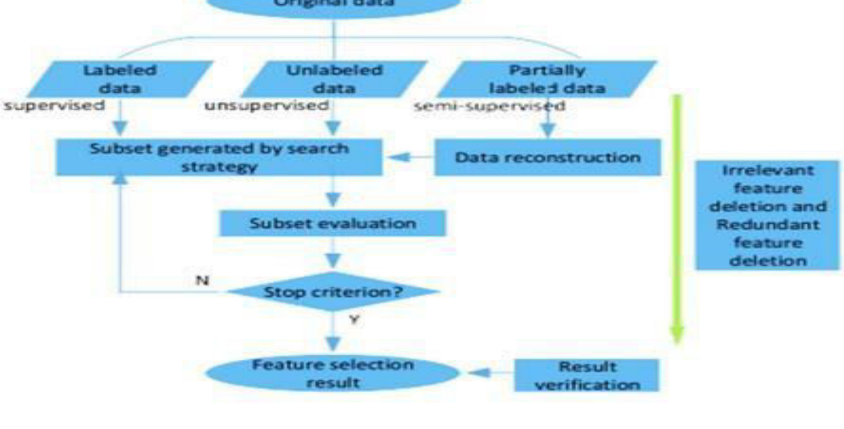


Figure 1: System Architecture.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. LITERATURE REVIEW

Many researchers have worked on predicting software defects using different techniques. Earlier approaches mainly focused on simple rule-based systems or basic machine learning models, which were useful but had several limitations, such as low accuracy and difficulty in handling large amounts of data.

For example, **H. Wang et al. (2015)** introduced a method to prioritize software testing by analyzing the impact of changes in code. This helped in finding defects faster. Similarly, **K. Zhai et al. (2014)** developed a method for testing location-based services, which improved the accuracy of defection detection for apps that rely on location data.

A study by **M. Bozkurt et al. (2013)** provided an overview of various techniques used in software testing, highlighting the challenges in automating the process. **Y. Ni et al. (2013)** focused on improving test case generation for certain software systems, making the process of detecting bugs more efficient. Later, **H. Wang et al. (2017)** introduced a smarter way to choose test cases, reducing the number of unnecessary tests while still catching software defects.

While these studies improved defect prediction and software testing, most of them focused only on detecting whether a defect exists or not, rather than identifying what type of defect it is. Additionally, they did not fully utilize natural language processing (NLP), which can analyze the text in bug reports to provide deeper insights.

This research builds on previous work by using NLP and ensemble machine learning models to classify software defects into multiple categories. By testing the model on real-world bug report datasets from Mozilla and Eclipse, this study aims to improve accuracy, reduce manual effort, and create a more efficient system for predicting software defects. While these studies contributed significantly to defect prediction and software testing, most of them focused on binary classification or regression testing without addressing the multi-class classification problem in bug reports. Additionally, many existing approaches lack the integration of natural language processing (NLP), which can extract valuable insights from textual bug reports to improve prediction accuracy. To address these limitations, this research builds upon previous studies by incorporating ensemble learning techniques and NLP-based feature extraction for multi-class defect prediction. By leveraging publicly available datasets from Mozilla and Eclipse, this study aims to enhance software defect classification accuracy while reducing manual effort and improving scalability.

III. METHODOLOGY OF PROPOSED SURVEY

The Software Development Life Cycle (SDLC) is a series of stages that provide a structured approach to the software development process. It encompasses understanding the business requirements, eliciting needs, converting concepts into functionalities and features, and ultimately delivering a product that meets business needs. A proficient software developer should possess adequate knowledge to select the appropriate SDLC model based on project context and business requirements. Therefore, it is essential to select the right SDLC model tailored to the specific concerns and requirements of the project to ensure its success. To explore more about choosing the right SDLC model, you can follow this link for additional information. Furthermore, to delve deeper into software lifecycle testing and SDLC stages, follow the highlighted links here. The exploration will cover various types of SDLC models, their benefits, disadvantages, and when to use them. SDLC models can be viewed as tools to enhance product delivery. Therefore, understanding each model, its advantages, disadvantages, and the appropriate usage is crucial to determine which one suits the project context.

Types of Software developing life cycles (SDLC)

- Waterfall Model
- V-Shaped Model
- Evolutionary Prototyping Model
- Spiral Method (SDM)
- Iterative and Incremental Method
- Agile development

IV.CONCLUSION AND FUTURE WORK

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Software defect prediction is a crucial area of research in software engineering, aiming to identify flaws in source code before testing. Traditional methods like system and unit testing become challenging as projects grow and complexity. This study analyzed five NASA datasets (JM1, CM1, KC1, KC2, and PC1) using machine learning techniques such as Bayesian Net, Logistic Regression, and Random Forest, paired with feature selection in WEKA. Future enhancements for this study could include integrating advanced deep learning models, like CNNs and LSTMs, to handle complex patterns in source code. Hybrid and ensemble approaches, combining algorithms such as Random Forest with boosting methods, could improve prediction accuracy and robustness.

REFERENCES

1. [Eletricity Information Sharing and Analysis Center (E-ISAC). (Mar. 2016). Analysis of the Cyber Attack on the Ukrainian Power Grid, Electricity Information Sharing and Analysis Center (E-ISAC), [Online]. Available: https://nsarchive.gwu.edu/sites/default/files/documents/3891751/SANS-and-Electricity-Information-Sharing-and.pdf
2. H. Zhang, B. Liu, and H. Wu, ‘‘Smart grid cyber-physical attack and defense: A review,’’ IEEE Access, vol. 9, pp. 29641–29659, 2021.
3. Gusrialdi and Z. Qu, ‘‘Smart grid security: Attacks and defenses,’’ in Smart Grid Control (Power Electronics and Power Systems), 1st ed. Cham, Switzerland: Springer, 2018, pp. 199–223.
4. R. Deng, P. Zhuang, and H. Liang, ‘‘False data injection attacks against state estimation in power distribution systems,’’ IEEE Trans. Smart Grid, vol. 10, no. 3, pp. 2871–2881, May 2019.
5. S. Lakshminarayana, J. Ospina, and C. Konstantinou, ‘‘Load-altering attacks against power grids under COVID-19 low-inertia conditions,’’ IEEE Open Access J. Power Energy, vol. 9, pp. 226–240, 2022.
6. I.-S. Choi, J. Hong, and T.-W. Kim, ‘‘Multi-agent based cyber attack detection and mitigation for distribution automation system,’’ IEEE Access, vol. 8, pp. 183495–183504, 2020.
7. J. Appiah-Kubi and C.-C. Liu, ‘‘Decentralized intrusion prevention (DIP) against co-ordinated cyberattacks on distribution automation systems,’’ IEEE Open Access J. Power Energy, vol. 7, pp. 389–402, 2020.
8. Moya and J. Wang, ‘‘Developing correlation indices to identify coordinated cyber-attacks on power grids,’’ IET Cyber-Phys. Syst., Theory Appl.,vol. 3, no. 4, pp. 178–186, Dec. 2018.
9. Y. Lin and Z. Bie, ‘‘Tri-level optimal hardening plan for a resilient distribution system considering reconfiguration and DG islanding,’’ Appl. Energy,vol. 210, pp. 1266–1279, Jan. 2018.
10. K. Lai, M. Illindala, and K. Subramaniam, ‘‘A tri-level optimization model to mitigate coordinated attacks on electric power systems in a cyberphysical environment,’’ Appl. Energy, vol. 235, pp. 204–218, Feb. 2019.
11. Abedi, M. R. Hesamzadeh, and F. Romerio, ‘‘An ACOPF-based bileveloptimization approach for vulnerability assessment of a power system,’’ Int. J. Electr. Power Energy Syst., vol. 125, Feb. 2021, Art. no. 106455.
12. L. Motto, J. M. Arroyo, and F. D. Galiana, ‘‘A mixed-integer LP procedure for the analysis of electric grid security under disruptive threat,’’ IEEE Trans. Power Syst., vol. 20, no. 3, pp. 1357–1365, Aug. 2005.
13. E. Shannon, ‘‘A mathematical theory of communication,’’ Bell Syst. Tech. J., vol. 27, no. 3, pp. 379–423, 1948.
14. J. R. Quinlan, ‘‘Induction of decision trees,’’ Mach. Learn., vol. 1, no. 1,pp. 81–106, Mar. 1986.W. Wu, B. Li, L. Chen, C. Zhang, and P. S. Yu, ‘‘Improved consistent weighted sampling revisited,’’ IEEE Trans. Knowl. Data Eng., vol. 31,no. 12, pp. 2332–2345, Dec. 2019.
15. R. S. Sutton and A. G. Barto, ‘‘Multi-armed bandits,’’ in Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018, ch. 2, pp. 32–33.
16. L. Gan and S. H. Low, ‘‘Convex relaxations and linear approximation for optimal power flow in multiphase radial networks,’’ in Proc.Power Syst. Comput. Conf., Aug. 2014, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/7038399
17. J. A. Taylor and F. S. Hover, ‘‘Convex models of distribution system reconfiguration,’’ IEEE Trans. Power Syst., vol. 27, no. 3, pp. 1407–1413, Aug. 2012.
18. Akyol, J. N. Haack, S. Ciraci, B. J. Carpenter, M. Vlachopoulou, and C. W. Tews. (Jun. 2012).

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com