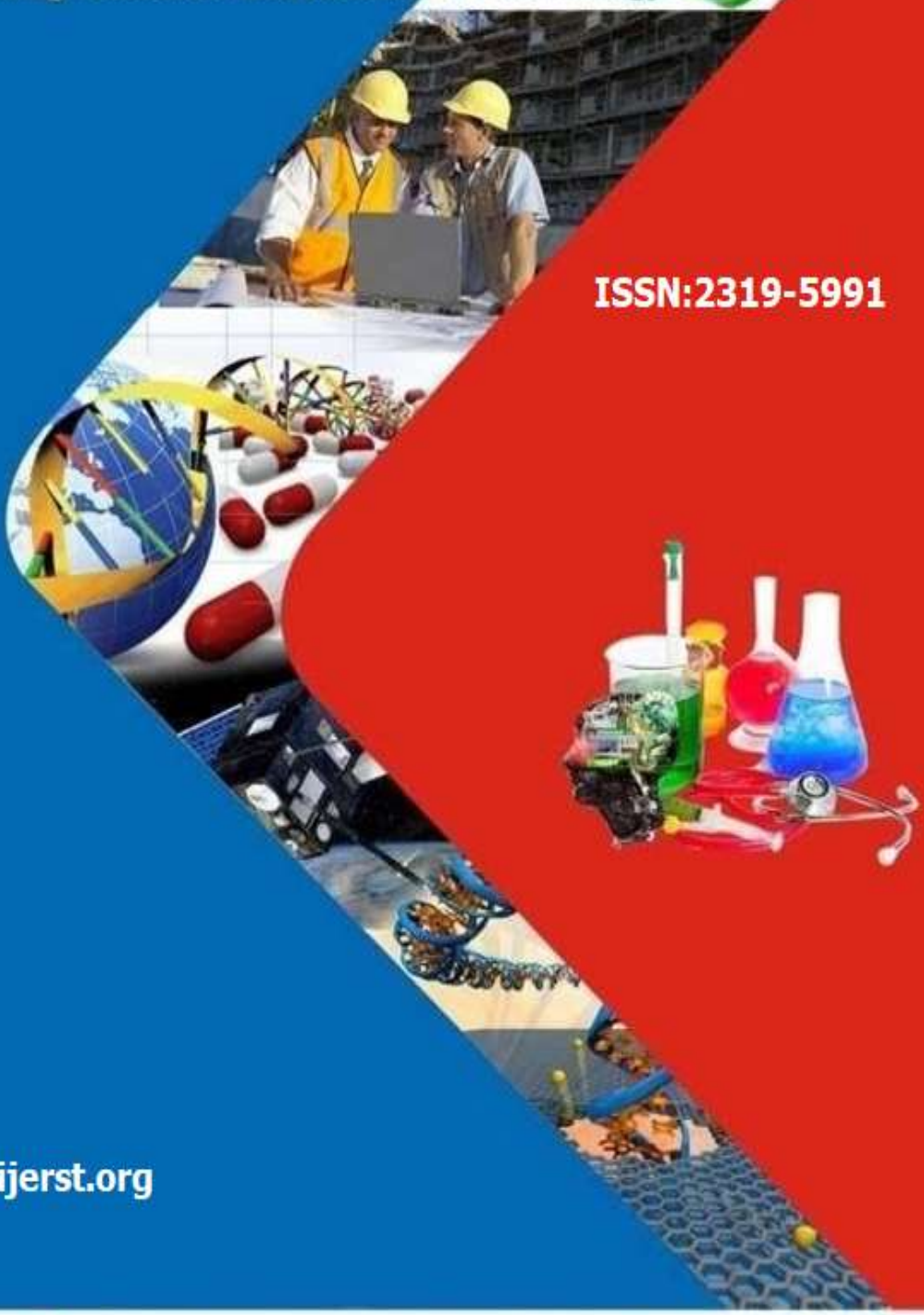
International Journal of Engineering Research and Science & Technology

IJERST

ISSN:2319-5991



www.ijerst.org

E-mail: editor@ijerst.org or ijerst.editor@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

PREDICTING BUG REPORTS WITH A NATURE-INSPIRED ENSEMBLE MACHINE LEARNING APPROACH

Mr. Nuthalapati Kiran Kumar, M.Tech
Assistant Professor, HOD ,
Department of MCA
Bapatla Engineering College
Bapatla, Andhra Pradesh

Manne Sai Ramya
Department of MCA
Bapatla Engineering College
Bapatla,Andhra Pradesh

Bonigala Jaya Rao
Department of MCA
Bapatla Engineering College
Bapatla,Andhra Pradesh

Tiragabathini Revanth Kumar
Department of MCA
Bapatla Engineering College
Bapatla,Andhra Pradesh

Nadakuditi Bharath
Department of MCA
Bapatla Engineering College
Bapatla,Andhra Pradesh

ABSTRACT

Researchers have focused on the maintenance process of software systems in software development systems because of its significance in resolving defects found during software testing through the use of bug reports (BRs), which contain comprehensive information such as the bug's description, status, reporter, assignee, priority, and severity, among other details. The primary challenge in this approach is figuring out how to examine these BRs in order to find every flaw in the system. Since there are a lot more BRs, doing this by hand is laborious and time-consuming. The automated solution is thus the most effective. The majority of current research is on automating this process from various angles, such determining the bug's importance or severity. They did not, however, take into account the fact that the defect is a multi-class categorisation issue. By putting out a novel prediction model to examine BRs and forecast the kind of defect, this study resolves this issue. Using machine learning and natural language processing (NLP) methods, the suggested model builds an ensemble machine learning algorithm. A publicly accessible dataset for two online software bug repositories (Eclipse and Mozilla) including six classes—Program Anomaly, GUI, Network or Security, Configuration, Performance, and Test-Code—is used to replicate the suggested approach. According to the simulation findings, the suggested model can outperform the majority of current models in terms of accuracy, achieving 90.42% without text augmentation and 96.72% with text augmentation..

1. INTRODUCTION

Testing is the assessment procedure used in software engineering to determine if a particular system satisfies the criteria and focusses on identifying errors or shortcomings in fulfilling these stakeholder-defined objectives [1]. Because of this procedure, flaws found after the testing

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

phase is complete are fixed during the maintenance phase. Software projects are more likely to have faults, and software developers prefer to deliver their products with flaws as software becomes bigger and more complicated [2]. As a result, users report these issues and flaws [2]. A bug is a defect, mistake, failure, or fault in software that causes it to perform improperly or provide inaccurate results [3]. The bug tracking system (BTS) receives the reporter's comments as a bug report. An example of a bug report in the Eclipse repository is shown in Figure 1.One Information regarding the found bug, including the bug ID, bug status (closed or opened), bug description, program information, component impacted, how to replicate the problem, bug reporter, and developer who fixed it, is included in a bug report [4].

A bug report can be thought as the medium that transfers and delivers the bug to the developers [5]. The process used by the developer after the assignment of the bug report to resolve it is the bug management process [6]. This process starts when this report is submitted by users to the bug management system when they face an error in a released software product. Then, this bug report is assigned to developers who work to find the location of the bug. The bug is fixed by the developer who finds the cause of the bug and its location before other developers. After the bug resolution, the tester checks the bug scenario, and if it does not reoccur, updates the bug report status to Verified. Finally, the reporter receivesa notification [6].

A software bug has its own life cycle made of different phases in its life. Therefore, Figure 2 illustrates the bug report life cycle. This figure shows that the life cycle has three phases, bug management, bug triage, and bug localization. The bug management phase includes all activities which start when users report the bug until the assignment of the developer. The next phase is bug triage when submitted reports are prioritized and assigned to a suitable developer for repair. Finally, the bug localization phase changes the bug status from resolved to verified to closed [7]. In this life cycle, the main challenge is the dramatically increasing number of bug reports, which are tedious, cumbersome, and time-consuming to manage manually [8]. Researchers address this issue by analyzing and classifying these reports according to three major categories, each with its sub-categories [9], and extracting useful information to accelerate and facilitate the maintenance phase. These categories of bug reports are classified by nature, priority, and severity [9]. Most studies classify bug reports by severity or priority.

The literature review in this research shows that various classification algorithms classify bug reports from different aspects, but there is a lack of nature-based bug classification models with high accuracy. Therefore, this paper intends to bridge this gap by introducing an ensemble machine

International Journal of Engineering Research and Science & Technology

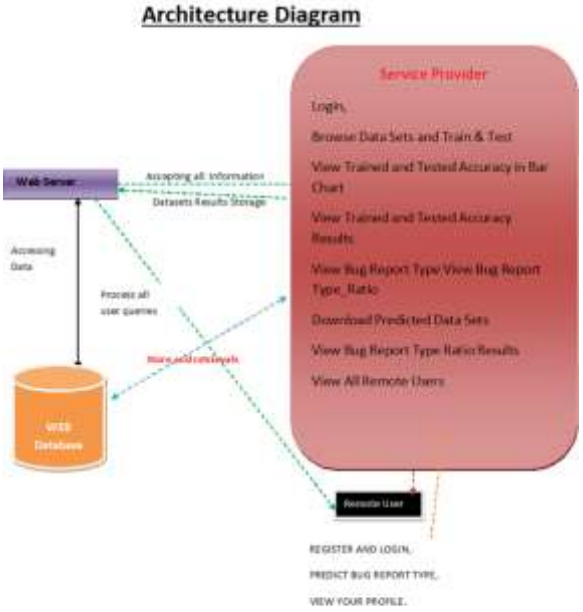
ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

learning algorithm for nature-based bug prediction from bug reports.

This research aims to provide a solution to maintaining software systems and contribute to this context by applying natural language processing (NLP), machine learning (ML), and text mining techniques to predict the bug types automatically. Once the model identifies the bug type, it accelerates the maintenance phase with bug localization techniques rather than doing this time-consuming task manually. Figure 3 illustrates an overview of the bug prediction process from a bug report.

II. SYSTEM ARCHITECTURE



III. EXISTING SYSTEM

Kukkar and Mohana [10] combine text mining, natural language processing, and machine learning techniques to classify bug and non-bug reports to solve the problem of misclassification of bug reports, which negatively affects the overall performance of the prediction process. This model uses bigram and TF-IDF in feature selection. However, the performance of the KNN algorithm in this model changes when the data-set changes.

Some researchers have different classifications for bug reports. In [11], researchers use machine learning algorithms to classify new bug reports as either corrective (defect fixing) reports or perfective (major maintenance) reports by using support vector machines (SVM), naive Bayes (NB), and random trees (RT). The results show that SVM achieved the highest accuracy, 93.1%.

To support the software engineering process, researchers have introduced a well-known schema called Orthogonal Defect Classification (ODC) [12]. ODC has eight orthogonal attributes to characterize software defects and several analytical methods for test process analysis and software development [51]. Extracting valuable information from defects can be done by ODC, which provides insights and helps diagnosis in software engineering processes [13]. The authors in [13] use machine learning algorithms to classify bug reports according to ODC. Their study uses 4096 ODC annotated bug reports. However, they conclude that there is difficulty in

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

automating the ODC attributes using only bug reports.

Hirsch and Hofer [14] classified each new bug according to a bug report using three classes, concurrency, memory, and semantic bugs. They used 369 bug reports, and the highest mean precision and recall were 0.74 and 0.72, respectively.

Disadvantages

- An existing system is not hybrid deep learning detection policy to improve the efficiency and effectiveness of Bug Report Generation.
- An existing system never used Nature-Based Ensemble Machine Learning Bug Prediction Model which is more accurate and efficient.

IV. PROPOSED SYSTEM

• It presents an evaluation of some base machine learning algorithms for an automatic nature-based bug report classification.
• It proposes an automated ensemble machine learning based approach for nature-based bug prediction from bug reports.
• It introduces a text augmentation technique in nature-based bug prediction from bug reports in the proposed model. To the best of our knowledge, it is the first ensemble machine learning approach to predict the nature of bugs in bug reports using a text augmentation technique.
• The evaluation results of the proposed approach on a benchmark dataset suggest that the proposed ensemble machine learning approach is accurate in nature-based bug prediction from bug reports.

Advantages

- The proposed algorithm aims to enhance nature-based bug prediction by using several machine learning (ML) base classifiers and training them using a benchmark dataset.

- The proposed algorithm integrates machine learning, NLP, and text-mining techniques. Their algorithm uses 504 bug reports and classifies them into four classes: assignment/initialization, external interface, internal interface, and other. This model achieved 73.70% of accuracy.

V. IMPLEMENTATION

Modules description

Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Browse Data Sets and Train & Test ,View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Bug Report Type, View Bug Report Type_Ratio,Download Predicted Data Sets ,View Bug Report Type Ratio Results, View All Remote Users.

View and Authorize Users

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT BUG REPORT TYPE, VIEW YOUR PROFILE.

VI. CONCLUSION

This paper proposed a nature-based bug prediction component using an ensemble machine learning algorithm that consists of four base machine learning algorithms, Random Forest, Support Vector Classification, Logistic Regression, and Multinomial Naïve Bayes. The accuracy of the model is 90.42%. Moreover, it utilizes a text augmentation technique to increase accuracy. Therefore, the highest accuracy achieved by the proposed model increased to 96.72%. The proposed model predicts the nature of the bug from six bug categories, Program Anomaly, GUI, Network or Security, Configuration, Performance, and Test-Code. Future work will enhance this model by increasing the number of bug categories and recommending possible solutions for predicted bugs to reduce the maintenance time.

REFERENCES

[1] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, ''Software testing techniques: A literature review,'' in Proc. 6th Int. Conf. Inf. Commun. Technol. Muslim World (ICT4M), Nov. 2016, pp. 177–182.

[2] W. Y. Ramay, Q. Umer, X. C. Yin, C. Zhu, and I. Illahi, ''Deep neural network-based severity prediction of bug reports,'' IEEE Access, vol. 7, pp. 46846–46857, 2019.

[3] W. Wen, ''Using natural language processing and machine learning techniques to characterize configuration bug reports: A study,'' M.S. thesis, College Eng., Univ. Kentucky, Lexington, KY, USA, 2017.

[4] J. Polpinij, ''A method of non-bug report identification from bug report repository,'' Artif. Life Robot., vol. 26, no. 3, pp. 318–328, Aug. 2021.

[5] S. Adhikarla, ''Automated bug classification.: Bug report routing,''
M.S. thesis, Fac. Arts Sci., Dept. Comput. Inf. Sci., Linköping Univ., Linköping, Sweden, 2020.

[6] K. C. Youm, J. Ahn, and E. Lee, ''Improved bug localization based on code change histories and bug reports,'' Inf. Softw. Technol., vol. 82, pp. 177–192, Feb. 2017.

[7] N. Safdari, H. Alrubaye, W. Aljedaani, B. B. Baez, A. DiStasi, and M. W. Mkaouer, ''Learning to rank faulty source files for dependent bug reports,'' in Proc. SPIE, vol. 10989, 2019, Art. no. 109890B.

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

[8] A. Kukkar, R. Mohana, A. Nayyar, J. Kim, B.-G. Kang, and N. Chilamkurti, ‘‘A novel deep-learning-based bug severity classification technique using convolutional neural networks and random forest with boosting,’’ Sensors, vol. 19, no. 13, p. 2964, Jul. 2019.

[9] A. Aggarwal. (May 2020). Types of Bugs in Software
Testing: 3 Classifications With Examples. [Online]. Available: https://www.scnsoft.com/software-testing/types-of-bugs

[10] A. Kukkar and R. Mohana, ‘‘A supervised bug report classification with incorporate and textual field knowledge,’’ Proc. Comput. Sci., vol. 132, pp. 352–361, Jan. 2018.

[11] A. F. Otoom, S. Al-jdaeh, and M. Hammad, ‘‘Automated classification of software bug reports,’’ in Proc. 9th Int. Conf. Inf. Commun. Manage., Aug. 2019, pp. 17–21.

[12] P. J. Morrison, R. Pandita, X. Xiao, R. Chillarege, and L. Williams, ‘‘Are vulnerabilities discovered and resolved like other defects?’’ Empirical Softw. Eng., vol. 23, no. 3, pp. 1383–1421, Jun. 2018.

[13] F. Lopes, J. Agnelo, C. A. Teixeira, N. Laranjeiro, and J. Bernardino, ‘‘Automating orthogonal defect classification using machine learning algorithms,’’ Future Gener. Comput. Syst., vol. 102, pp. 932–947, Jan. 2020.

[14] T. Hirsch and B. Hofer, ‘‘Root cause prediction based on bug reports,’’ in Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW), Oct. 2020, pp. 171–176.

[15] Q. Umer, H. Liu, and I. Illahi, ‘‘CNN-based automatic prioritization of bug reports,’’ IEEE Trans. Rel., vol. 69, no. 4, pp. 1341–1354, Dec. 2020.