International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 8, May 2026
Impact Factor: 8.2

# A Novel Method to Enhance the Accuracy of Software for Predicting Defects Using Machine Learning

**M. Maruthi Rao[1] and Smt. D. Shoba Rani[2]**
[1]PG Scholar, Department of CSE
[2]Head & Professor, Department of CSE
Chadalawada Ramanamma Engineering College (Autonomous), Tirupati.

**Abstract:** *In the software engineering community, forecasting defects is a highly active domain. For the success of a software system, it is crucial to bridge the gap between software engineering and data mining techniques. Software defect prediction aims to identify source code errors before the testing phase commences. Various methods are commonly employed to explore the affected areas within software, such as clustering, statistical approaches, hybrid algorithms. The primary contribution of this research is the novel application of feature selection to enhance the accuracy of machine learning classifiers in defect prediction. The results of this study reveal that the accuracy of defect prediction with feature selection is improved in contrast with the accuracy achieved without feature selection.*

**Keywords:** Accuracy, Classifiers, Defects, Machine learning. Software

## I. INTRODUCTION

Software defects are typically characterized as unforeseen or irregular behaviours displayed by a software system in response to user requirements or operational requests. These problems are frequently detected during the software testing phase, as testers note discrepancies or divergences from anticipated functionality. In software engineering, faults denote defects created throughout the software development process that may ultimately result in system failure or suboptimal performance from the user's perspective. An error generally arises from human errors in software design, coding, or implementation, while a defect denotes a discernible weakness in the software that results in erroneous consequences during execution.The discovery and prediction of software flaws entail analysing defective software modules and assessing various testing parameters. Creating an effective defect prediction model continues to pose a considerable problem in software engineering, especially for the early identification of faulty components in the software development life cycle. Conventional defect detection methodologies encompass source code review, unit testing, integration testing, beta testing, and system testing. As software systems expand in size, complexity, and code volume, testing operations become progressively more challenging and resource-demanding. As a result, software defect prediction has garnered significant attention in recent years because of its critical role in enhancing software quality. Defective modules can adversely impact the overall reliability of software products, leading to escalated development expenses, project delays, and augmented maintenance tasks.

## II. EXISTING SYSTEM

Software defect prediction has become a significant study area, especially through the utilization of machine learning algorithms, software metrics, and data-driven methodologies. Researchers have suggested many approaches and frameworks to enhance the precision of defect identification and prediction. Diverse methodologies have facilitated the creation of numerous prediction models and analyses intended to improve software reliability and quality. Between1990 and 2022, a large number of studies on software defect prediction analysis were published. In 1999,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 8, May 2026
Impact Factor: 8.2

Benton and Neil developed size and complexity measurements for defect prediction and talked about how crucial these metrics are to the prediction process. Defect prediction uses multivariate techniques, software measures, high-quality data processing, and critiques of existing approaches. Based on their calculations, there are for every thousand lines of code (KLOC), there are about 23 mistakes. Using neural networks to identify mistakes and machine learning (ML) techniques to forecast flaws. The study concluded that neural network-based strategies had enhanced efficacy in identifying software faults relative to alternative methods. The researchers assessed the efficacy of various machine learning algorithms by employing the PROMISE dataset to evaluate and compare their predicted performance. They proposed that key indicators of programming quality include lines of code (LOC), class reactivity, and the absence of effective coding strategies.

## III. PROPOSED SYSTEM

Software quality, dependability, and user happiness can all be severely harmed by software flaws. Conventional software defect prediction techniques frequently rely on code metrics, manual inspection, and historical data, which can be laborious, subjective, and inadequate for spotting intricate and dynamic flaws. More precise and effective methods for predicting software defects that use machine learning algorithms to examine software artefacts and spot possible flaws early in the development process are required program artefacts, including defect IDs, program names, and creation dates, will be gathered by the suggested method. In order to extract pertinent elements, such as code complexity metrics, code churn, developer expertise, and code modification trends, these data will be pre-processed. To get the data ready for machine learning modelling, methods for data cleaning, normalisation, and feature selection will be used.

## IV. DESIGN

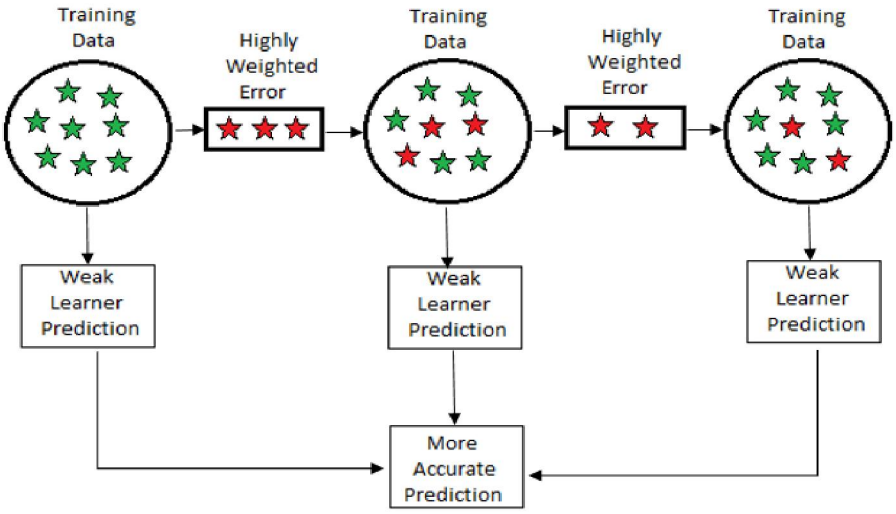


Fig:1 GBA Architecture

It is the procedure of creating the parts or modules of a new business system or modifying an existing one to satisfy the needs. Before making plans, we need to completely understand the previous system and choose the best ways to use computers.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 8, May 2026
Impact Factor: 8.2

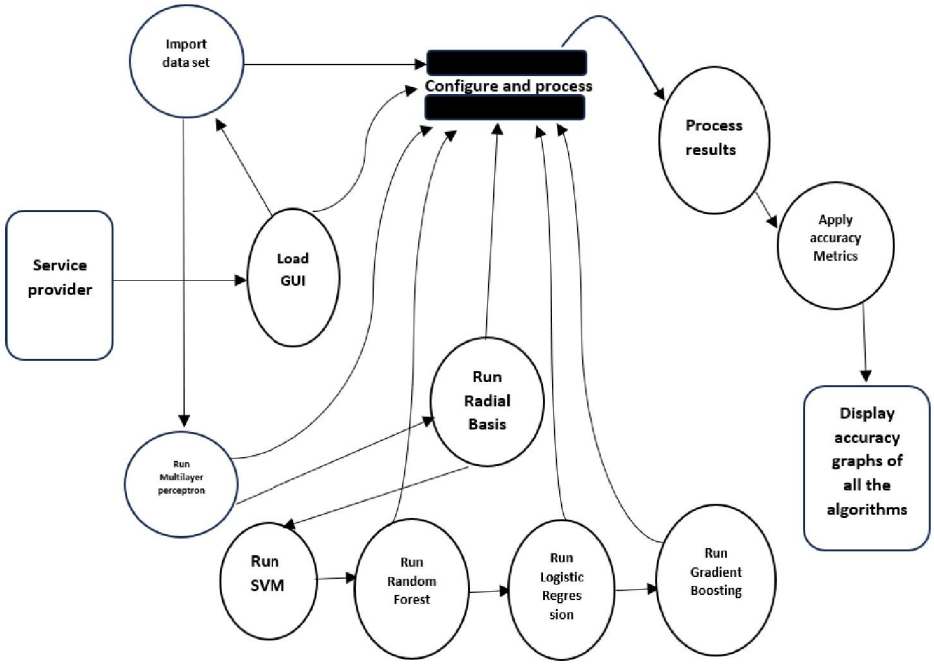


Fig:2 Data Flow Diagram

The Unified Modelling Language (UML) is a standardized modelling language employed for the design, visualisation, documentation, and specification of software systems, together with specific business and non-software processes. It offers a cohesive framework that facilitates the depiction of system architecture and behavior via a collection of graphical notations. UML integrates widely recognised software engineering methodologies that facilitate the effective modelling of large-scale and intricate systems.

## V. OO ANALYSIS AND DESIGN

A comprehensive grasp of object-oriented (OO) analysis and design principles is crucial for efficient system development. The principal aim of object-oriented analysis is to discern the pertinent objects necessary for system architecture and to comprehend their functions within the application. This research can also be utilised to assess the structure and functionality of existing systems. Effective system analysis is achievable when the problem domain is scrutinized from an object-oriented viewpoint, facilitating the discovery of significant items and their interconnections. Upon recognizing the requisite objects and establishing their interrelationships, the system design may be formulated accordingly.

## VI. CONCLUSION

Software defect prediction is a significant and extensively studied domain within software engineering, owing to its contribution to improving software quality and minimizing development expenses. The principal aim of defect prediction is to detect potential errors in source code before to the software's testing phase, facilitating early remediation and reducing subsequent maintenance requirements. No singular prediction strategy is universally applicable to all datasets, as the efficacy of a method is predominantly contingent upon the qualities and quality of the underlying data. Thus, choosing an effective method for software defect prediction continues to pose a considerable difficulty in software engineering research.

International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 8, May 2026
Impact Factor: 8.2

## REFERENCES

**[1].** M. A. Memon, M.-U.-R. Magsi, M. Memon, and S. Hyder, ''Defects prediction and prevention approaches for quality software development,'' Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 8, pp. 451–457, 2018.

**[2].** M. Gayathri and A. Sudha, ''Software defect prediction system using multilayer perceptron neural network with data mining,'' Int. J. Recent Technol. Eng., vol. 3, no. 2, pp. 2277–3878, 2014.

**[3].** R. Malhotra, L. Bahl, S. Sehgal, and P. Priya, ''Empirical comparison of machine learning algorithms for bug prediction in open-source software,'' in Proc. Int. Conf. Big Data Anal. Comput. Intell. (ICBDAC), Andhra Pradesh, India, 2017, pp. 40–45, doi: 10.1109/ICBDACI.2017.8070806.

**[4].** M. S. Rawat and S. K. Dubey, ''Software defect prediction models for quality improvement: A literature study,'' Int. J. Comput. Sci. Issues, vol. 9, pp. 288–296, Jan. 2012.

**[5].** Singh, ''A survey? Data mining techniques in software engineering,'' Int. J. Res. IT, Manage. Eng., vol. 6, no. 3, pp. 30–34, Mar. 2016.

**[6].** N. Kalaivani and R. Beena, ''Overview of software defect prediction using machine learning algorithms,'' Int. J. Pure Appl. Math., vol. 118, pp. 3863–3873, Feb. 2018.

**[7].** M. Dhiauddin and S. Ibrahim, ''A prediction model for system testing defects using regression analysis,'' Int. J. Soft Comput. Softw. Eng., vol. 2, no. 7, pp. 55–68, Jul. 2012.

**[8].** R.Malhotraand A Jain, ''Fault prediction using statistical and machine learning methods for improving software quality,'' J. Inf. Process. Syst., vol. 8, no. 2, pp. 241–262, Jun. 2012.

**[9].** A Hammouri, M. Hammad, M. Alnabhan, and F. Alsarayrah, ''Software bug prediction using machine learning approach,'' Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 2, pp. 78–83, 2018.

**[10].** M. M. A. Abdallah and M. M. Alrifaee, ''Towards a new framework of program quality measurement based on programming language standards,'' Int. J. Eng. Technol., vol. 7, pp. 1–3, Oct. 2018