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 9, June 2026
Impact Factor: 8.2

An Enhanced Ensemble-Based Software Defect Prediction Framework Using Software Metrics and Machine Learning Techniques

Rajeev P R[1] and Sruthi R Varrier[2]
Assistant Professor, Department of Computer Applications[1]
Assistant Professor, Department of Electrical & Computer Engineering[2]
College of Engineering Kidangoor, Kottayam, India

Abstract: *Software defect prediction plays a vital role in software quality assurance by enabling the early identification of defect-prone software modules and facilitating efficient allocation of testing resources. The increasing complexity of modern software systems necessitates the development of accurate and reliable defect prediction techniques to reduce maintenance costs and improve software reliability. Traditional software defect prediction approaches often suffer from limitations related to noisy datasets, redundant software metrics, and the inability of individual classifiers to effectively capture complex relationships among software attributes.*

To address these challenges, this paper proposes an Enhanced Ensemble Software Defect Prediction (EESDP) framework that integrates software metrics, data preprocessing, feature ranking, and ensemble machine learning techniques for effective defect prediction. The proposed framework employs data cleaning and preprocessing mechanisms to improve dataset quality, followed by feature ranking to identify significant software metrics. Ensemble classification is achieved through the integration of Random Forest and Gradient Boosting classifiers using a voting-based decision mechanism. The framework is evaluated using benchmark NASA Metrics Data Program (MDP) datasets, including KC1, KC2, CM1, PC1, and JM1.

The proposed approach aims to improve prediction robustness by combining the strengths of multiple classifiers while reducing the influence of noisy and redundant attributes. The effectiveness of the proposed framework is discussed using standard performance measures such as Accuracy, Precision, Recall, and F1-Score. The framework provides a foundation for future experimental validation using benchmark software defect prediction datasets. The proposed EESDP framework provides a practical and scalable solution for software defect prediction and contributes to the development of reliable software systems..

Keywords: Software Defect Prediction, Software Metrics, Ensemble Learning, Random Forest, Gradient Boosting, Software Quality Assurance, NASA MDP Datasets

I. INTRODUCTION

Software quality is one of the most significant concerns in modern software engineering due to the increasing complexity, scale, and functionality of software systems. Software defects introduced during the software development life cycle can adversely affect software reliability, maintainability, security, and user satisfaction. The presence of defects often leads to increased development costs, delayed project completion, system failures, and significant maintenance efforts. Therefore, early identification of defect-prone software modules has become an important research area in software quality assurance.

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 9, June 2026 Impact Factor: 8.2

Software Defect Prediction (SDP) aims to identify software components that are likely to contain defects before deployment. Effective defect prediction enables software organizations to focus testing resources on high-risk modules, thereby reducing maintenance costs and improving software quality. Early detection of software defects not only minimizes project risks but also enhances the reliability and robustness of software products.

Software metrics have been widely used as indicators for software defect prediction. Metrics such as Lines of Code (LOC), Cyclomatic Complexity, Essential Complexity, Design Complexity, and Halstead Metrics provide quantitative measurements of software characteristics and complexity. These metrics help software engineers understand the structural properties of software modules and identify fault-prone components. Benchmark datasets such as the NASA Metrics Data Program (MDP) datasets have become popular resources for evaluating software defect prediction techniques because they contain software metrics and corresponding defect labels collected from real-world software projects.

Traditional software defect prediction approaches primarily utilize statistical methods and machine learning algorithms such as Decision Trees, Naïve Bayes, Logistic Regression, Support Vector Machines, and k-Nearest Neighbors. Although these techniques have demonstrated promising results, their prediction performance is often influenced by noisy datasets, redundant attributes, class imbalance, and limited capability to model complex relationships among software metrics. Furthermore, individual classifiers may perform inconsistently across different software projects and datasets.

Recent advances in machine learning have introduced ensemble learning techniques that combine multiple classifiers to improve prediction performance. Ensemble learning leverages the strengths of individual classifiers while minimizing their weaknesses through collaborative decision-making. Methods such as Random Forest, Gradient Boosting, AdaBoost, and Voting Ensembles have demonstrated superior classification performance in various application domains. By integrating multiple learning models, ensemble approaches can improve robustness, reduce overfitting, and enhance generalization capability.

Despite significant progress in software defect prediction research, several challenges remain unresolved. Software metrics datasets frequently contain noisy records, missing values, and redundant attributes that negatively affect prediction accuracy. In addition, many existing studies focus on individual classifiers and do not fully exploit the advantages of ensemble learning strategies. Therefore, there is a need for an integrated framework that combines effective preprocessing, feature ranking, and ensemble classification techniques to improve software defect prediction performance.

To address these challenges, this paper proposes an Enhanced Ensemble Software Defect Prediction (EESDP) framework that integrates software metrics, data preprocessing, feature ranking, and ensemble machine learning techniques. The proposed framework employs preprocessing mechanisms to improve dataset quality, applies feature ranking to identify significant software metrics, and utilizes a voting-based ensemble model that combines Random Forest and Gradient Boosting classifiers for defect prediction. The framework is evaluated using benchmark NASA MDP datasets, including KC1, KC2, CM1, PC1, and JM1.

The major contributions of this research are summarized as follows:

• Development of an Enhanced Ensemble Software Defect Prediction (EESDP) framework for software quality assessment.

• Integration of data preprocessing and feature ranking techniques to improve software metrics quality.

• Application of ensemble learning techniques for effective identification of defect-prone software modules.

• Evaluation of the proposed framework using benchmark NASA MDP software defect datasets.

• Provision of a practical and scalable solution for software quality assurance and maintenance planning.

The remainder of this paper is organized as follows. Section 2 reviews existing research on software defect prediction techniques. Section 3 presents the proposed EESDP framework and methodology. Section 4 describes the experimental setup and dataset analysis. Section 5 discusses the results and performance evaluation. Finally, Section 6 concludes the paper and outlines future research directions.

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 9, June 2026
Impact Factor: 8.2

II. RELATED WORK

Software defect prediction has attracted significant attention in software engineering due to its ability to identify defect-prone modules during the early stages of software development. Accurate prediction models help software organizations reduce maintenance costs, improve software quality, and optimize testing efforts. Over the years, researchers have proposed numerous techniques based on software metrics, machine learning, evolutionary algorithms, and deep learning methods to improve defect prediction performance.

Early software defect prediction studies primarily utilized software metrics such as Lines of Code (LOC), McCabe Cyclomatic Complexity, and Halstead Metrics to identify fault-prone software modules. These metrics provide quantitative measurements of software complexity, maintainability, and reliability, enabling the development of predictive models based on historical software data. Statistical techniques and traditional classification algorithms such as Logistic Regression, Decision Trees, and Naïve Bayes were widely adopted for software defect prediction due to their simplicity and ease of implementation.

Machine learning approaches have significantly improved the effectiveness of software defect prediction systems. Decision Tree classifiers have been extensively used because of their interpretability and ability to generate understandable prediction rules. Naïve Bayes classifiers have demonstrated efficient performance in handling large software datasets and probabilistic classification tasks. Similarly, Support Vector Machines (SVM) have shown promising results by effectively separating defective and non-defective software modules through hyperplane-based classification mechanisms. Random Forest classifiers have gained popularity because they combine multiple decision trees and provide improved classification accuracy while reducing overfitting problems.

Researchers have also explored feature selection and optimization techniques to improve prediction performance. Evolutionary algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) have been employed to identify optimal feature subsets from software metrics datasets. These approaches help eliminate redundant attributes, reduce computational complexity, and improve classification efficiency. Several studies reported that combining optimization algorithms with machine learning classifiers significantly enhances software defect prediction accuracy.

Rule mining approaches have also been investigated for software defect prediction. Association rule mining techniques identify hidden relationships among software metrics and generate interpretable prediction rules. Rule-based models provide transparency in decision-making and enable software engineers to understand the factors contributing to software defects. However, traditional rule mining approaches often generate a large number of redundant rules, resulting in increased computational complexity and reduced prediction efficiency.

In recent years, deep learning techniques have emerged as powerful tools for software defect prediction. Artificial Neural Networks (ANN), Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Autoencoders have been applied to learn complex relationships among software metrics and defect labels. Deep learning models have demonstrated improved prediction capability compared with traditional machine learning approaches, particularly when large datasets are available. Despite their effectiveness, deep learning models often require extensive computational resources and are frequently considered black-box systems due to their limited interpretability.

Ensemble learning has recently gained attention as an effective approach for improving classification performance. Ensemble techniques combine the predictions of multiple classifiers to generate a more robust and accurate prediction model. Random Forest, Gradient Boosting, AdaBoost, and Voting-based classifiers have demonstrated superior performance in various software engineering applications. By leveraging the strengths of multiple learning algorithms, ensemble methods can improve prediction accuracy, reduce variance, and enhance generalization capability.

Although considerable progress has been made in software defect prediction research, several challenges remain unresolved. Software metrics datasets frequently contain noisy records, missing values, and redundant attributes that negatively impact classification performance. Many existing approaches focus on a single classification technique and fail to fully exploit the benefits of ensemble learning. Furthermore, the integration of effective preprocessing, feature ranking, and ensemble classification within a unified framework remains relatively unexplored.

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 9, June 2026
Impact Factor: 8.2

To address these limitations, this research proposes an Enhanced Ensemble Software Defect Prediction (EESDP) framework that combines data preprocessing, feature ranking, and ensemble machine learning techniques for improved software defect prediction. The proposed framework aims to provide a reliable, scalable, and cost-effective solution for identifying defect-prone software modules and supporting software quality assurance activities.

2.1 Research Gap

A comprehensive review of existing software defect prediction techniques reveals several limitations. Traditional machine learning approaches often suffer from reduced prediction accuracy when handling noisy and high-dimensional software metrics datasets. Optimization-based approaches improve feature selection but increase computational complexity. Rule mining techniques provide interpretability; however, they frequently generate redundant rules and require extensive post-processing. Deep learning approaches achieve high prediction accuracy but often lack transparency and demand significant computational resources.

Furthermore, most existing studies focus on individual prediction techniques rather than leveraging the collective strengths of multiple classifiers. Limited research has investigated the integration of preprocessing, feature ranking, and ensemble learning within a unified software defect prediction framework. Therefore, there is a need for an enhanced ensemble-based framework capable of improving prediction performance while maintaining robustness and scalability. This research addresses this gap through the proposed Enhanced Ensemble Software Defect Prediction (EESDP) framework.

III. PROPOSED METHODOLOGY

3.1 Overview of the Proposed EESDP Framework

This research proposes an Enhanced Ensemble Software Defect Prediction (EESDP) framework that integrates software metrics, preprocessing techniques, feature ranking, and ensemble machine learning algorithms to improve software defect prediction performance. The framework is designed to identify defect-prone software modules at an early stage of software development, thereby reducing testing effort and maintenance cost while improving software quality.

The proposed EESDP framework is organized into five sequential phases, namely Dataset Collection, Data Preprocessing, Feature Ranking and Selection, Ensemble Classification, and Performance Evaluation. The integration of these phases enables systematic processing of software metrics and supports accurate prediction of software defects.

The overall workflow of the proposed framework is shown below:

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 9, June 2026
Impact Factor: 8.2

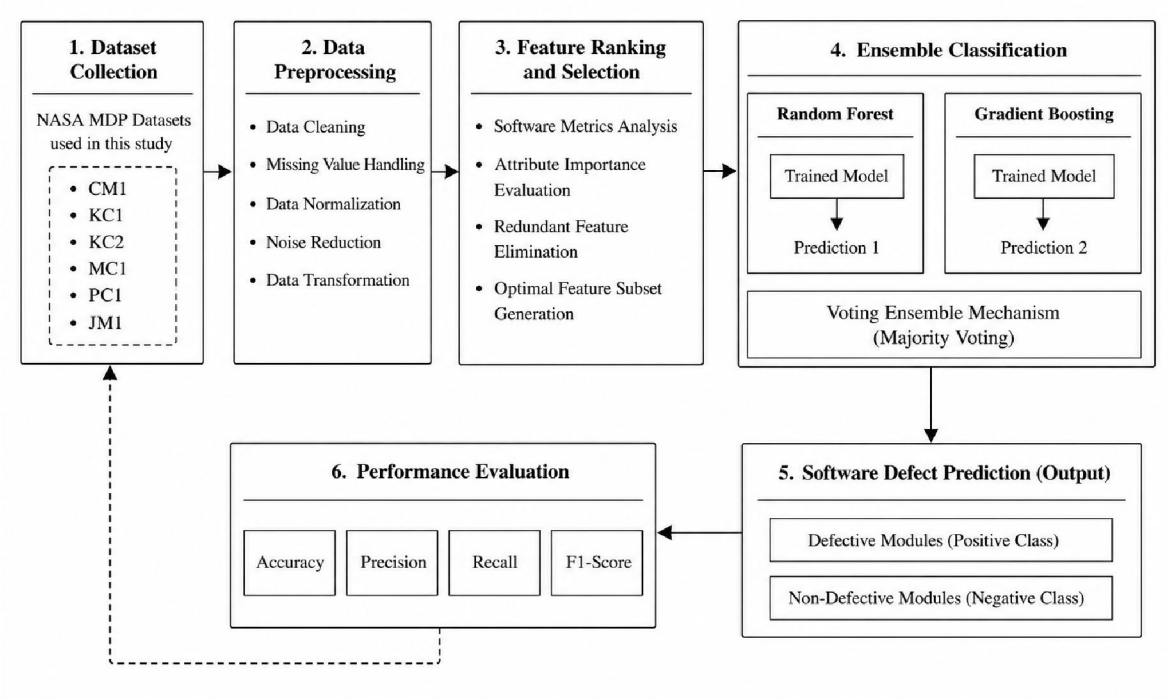

Figure 1. Architecture of the Proposed EESDP Framework

Figure 1 illustrates the overall architecture of the proposed EESDP framework. The framework integrates preprocessing, feature ranking, ensemble classification, and performance evaluation for effective software defect prediction.

3.2 Dataset Collection

The proposed framework utilizes benchmark software defect datasets obtained from the NASA Metrics Data Program (MDP). The selected datasets include KC1, KC2, CM1, PC1, and JM1, which are widely used in software defect prediction research.

The characteristics of the selected NASA MDP datasets are presented in Table 1.

These datasets contain software modules represented through various software metrics and corresponding defect labels indicating whether a module is defective or non-defective.

Dataset	Instances	Attributes
KC1	145	95
KC2	522	22
CM1	327	38
PC1	1109	22
JM1	10885	22

Table 1. NASA MDP Dataset Description

DOI: 10.48175/IJARSCT-37155

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 9, June 2026
Impact Factor: 8.2

As shown in Table 1, the selected datasets vary in size and attribute composition, providing a comprehensive environment for evaluating software defect prediction models.

3.3 Data Preprocessing

Software metrics datasets often contain inconsistencies, redundant attributes, and missing values that negatively affect classification performance. Therefore, a preprocessing phase is incorporated before model training.

The preprocessing stage includes:

Data Cleaning

Duplicate records and inconsistent entries are identified and removed to improve data quality.

Missing Value Handling

Missing attribute values are replaced using appropriate statistical techniques to maintain dataset completeness.

Data Normalization

The software metrics are normalized to a common numerical scale to prevent attributes with larger values from dominating the learning process.

These preprocessing activities improve the reliability and effectiveness of the subsequent classification process.

3.4 Feature Ranking and Selection

Software defect datasets frequently contain irrelevant and redundant attributes that increase computational complexity and reduce prediction performance.

Feature ranking is employed to evaluate the contribution of each software metric toward defect prediction. Metrics with higher significance scores are retained while less relevant attributes are eliminated.

The major software metrics considered for defect prediction are summarized in Table 2.

Metric Category	Metrics
Size Metrics	LOC
Complexity Metrics	Cyclomatic Complexity, Essential Complexity
Design Metrics	Design Complexity
Halstead Metrics	Volume, Difficulty, Effort, Program Level
Object-Oriented Metrics	CBO, RFC, WMC, DIT, NOC

Table 2. Software Metrics Used for Software Defect Prediction

Table 2 presents the major categories of software metrics used in this study. These metrics provide quantitative information regarding software size, complexity, maintainability, and fault proneness.

The objectives of feature ranking include:

- Reducing dimensionality.
- Improving classifier efficiency.
- Enhancing prediction accuracy.
- Minimizing overfitting.

The selected feature subset is subsequently used for classifier training and testing.

3.5 Ensemble Classification

The classification component of the EESDP framework integrates multiple machine learning models to improve prediction robustness.

A. Random Forest Classifier

Random Forest constructs multiple decision trees using random subsets of training data and features. The final prediction is obtained through majority voting among the individual trees.

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 9, June 2026
Impact Factor: 8.2

Advantages include:
• High classification accuracy.
• Reduced overfitting.
• Effective handling of high-dimensional datasets.

B. Gradient Boosting Classifier

Gradient Boosting sequentially builds weak learners that focus on correcting the errors of previous learners. This iterative learning mechanism improves predictive capability and reduces classification error.

Advantages include:
• Improved prediction performance.
• Effective handling of complex data relationships.
• High generalization capability.

C. Voting Ensemble

The outputs generated by Random Forest and Gradient Boosting classifiers are combined through a voting mechanism. The final class label is determined based on the majority prediction produced by the ensemble classifiers. This strategy improves prediction reliability and reduces the weaknesses associated with individual classifiers.

3.6 Performance Evaluation

The effectiveness of the proposed framework is assessed using standard classification measures. The performance evaluation metrics used in this study are summarized in Table 3.

Metric	Description
Accuracy	Overall prediction correctness
Precision	Correctness of positive predictions
Recall	Ability to identify defective modules
F1-Score	Harmonic mean of Precision and Recall

Table 3. Performance Evaluation Metrics

Table 3 presents the evaluation metrics used to assess the effectiveness of the proposed EESDP framework. These measures provide a comprehensive assessment of classification performance.

Accuracy

Accuracy measures the percentage of correctly classified software modules.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Precision

Precision measures the proportion of predicted defective modules that are actually defective.

$$\text{Precision} = TP / (TP + FP)$$

Recall

Recall measures the proportion of actual defective modules correctly identified.

$$\text{Recall} = TP / (TP + FN)$$

F1-Score

F1-Score provides a balanced measure of Precision and Recall.

$$\text{F1-Score} = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall})$$

These performance metrics provide a comprehensive evaluation of the software defect prediction capability of the proposed EESDP framework.

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 9, June 2026
Impact Factor: 8.2

IV. EXPERIMENTAL SETUP AND DATASET ANALYSIS

4.1 Experimental Environment

The proposed Enhanced Ensemble Software Defect Prediction (EESDP) framework was evaluated using benchmark software defect prediction datasets obtained from the NASA Metrics Data Program (MDP). The experimental framework was designed to assess the capability of ensemble learning techniques in identifying defect-prone software modules based on software metrics.

The experimental process consists of dataset collection, data preprocessing, feature ranking, classifier training, ensemble classification, and performance evaluation. The selected datasets contain software metrics and defect labels that enable binary classification of software modules into defective and non-defective categories.

4.2 Dataset Analysis

The proposed framework is designed for evaluation using five widely used NASA MDP datasets namely KC1, KC2, CM1, PC1, and JM1. These datasets contain software modules characterized by software metrics and corresponding defect information.

The datasets vary in terms of size, complexity, and attribute composition, thereby providing a comprehensive benchmark for evaluating software defect prediction models.

The characteristics of the selected datasets are presented in Table 4.

Dataset	Instances	Attributes
KC1	145	95
KC2	522	22
CM1	327	38
PC1	1109	22
JM1	10885	22

Table 4. Characteristics of NASA MDP Datasets

As shown in Table 4, JM1 represents the largest dataset, while KC1 contains the highest number of attributes. The diversity among the datasets enables the evaluation of the proposed framework under different software project conditions.

4.3 Class Distribution Analysis

Class distribution plays an important role in software defect prediction because highly imbalanced datasets can influence classification performance. The software modules in the NASA MDP datasets are categorized into defective and non-defective classes.

The distribution of software modules is analyzed before classifier training to understand dataset characteristics and support appropriate preprocessing operations.

The class labels used in the datasets are summarized in Table 5.

Dataset	Defective Class	Non-Defective Class
KC1	TRUE	FALSE
KC2	YES	NO
CM1	Y	N
PC1	TRUE	FALSE
JM1	TRUE	FALSE

Table 5. Defect Class Labels in NASA MDP Datasets

DOI: 10.48175/IJARSCT-37155

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 9, June 2026
Impact Factor: 8.2

The presence of binary defect labels enables the application of supervised machine learning algorithms for software defect prediction.

4.4 Experimental Procedure

The experimental procedure adopted in this study consists of the following steps:
Step 1: Collection of software metrics datasets.
Step 2: Data cleaning and preprocessing.
Step 3: Missing value handling and normalization.
Step 4: Feature ranking and selection.
Step 5: Training of Random Forest and Gradient Boosting classifiers.
Step 6: Ensemble prediction through voting mechanism.
Step 7: Evaluation using Accuracy, Precision, Recall, and F1-Score.
The experimental workflow adopted for the proposed framework is illustrated in Figure 2.

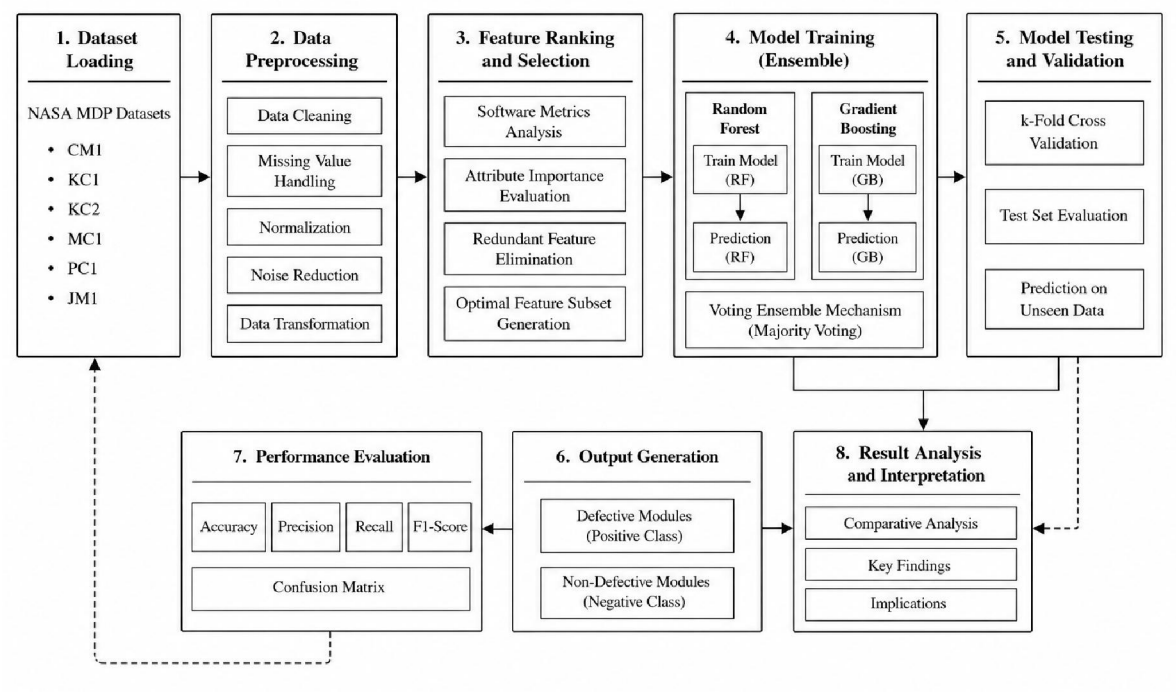

Figure 2. Experimental Workflow of the Proposed EESDP Framework

4.5 Evaluation Criteria

The effectiveness of the proposed EESDP framework is evaluated using standard classification performance measures. These measures provide insights into the prediction capability of the proposed ensemble model.
The evaluation measures considered in this study are presented in Table 6.

Metric	Purpose
Accuracy	Overall prediction performance
Precision	Correctness of positive predictions
Recall	Ability to identify defective modules
F1-Score	Balance between Precision and Recall

Table 6. Classification Performance Measures

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 9, June 2026
Impact Factor: 8.2

These metrics provide a comprehensive assessment of the effectiveness of the proposed software defect prediction framework.

V. DISCUSSION

The proposed Enhanced Ensemble Software Defect Prediction (EESDP) framework integrates software metrics, data preprocessing, feature ranking, and ensemble machine learning techniques to improve the effectiveness of software defect prediction. The framework is designed to address several challenges commonly encountered in traditional software defect prediction approaches, including noisy datasets, redundant attributes, class imbalance, and the limitations of individual classifiers.

The preprocessing component of the EESDP framework plays an important role in improving the quality of software metrics data. Software defect datasets often contain inconsistencies, missing values, and irrelevant information that negatively affect classification performance. The incorporation of data cleaning, missing value handling, normalization, and noise reduction techniques improves dataset quality and provides a reliable foundation for subsequent classification tasks.

Feature ranking and selection constitute another important component of the proposed framework. Software defect prediction datasets frequently contain a large number of software metrics, some of which may have limited contribution to defect identification. The feature ranking process evaluates the significance of software metrics and identifies the most relevant attributes for classification. By eliminating redundant and less informative features, the framework reduces computational complexity, improves learning efficiency, and minimizes the risk of overfitting.

The ensemble learning strategy employed in the proposed framework combines the strengths of Random Forest and Gradient Boosting classifiers through a voting-based mechanism. Random Forest provides robustness against overfitting and effectively handles high-dimensional software metrics datasets, while Gradient Boosting improves predictive capability by sequentially correcting classification errors. The voting ensemble mechanism integrates the predictions generated by both classifiers and produces a more reliable final decision regarding software defectiveness.

The NASA Metrics Data Program (MDP) datasets utilized in this study provide a suitable benchmark environment for software defect prediction research. The selected datasets, namely CM1, KC1, KC2, MC1, PC1, and JM1, contain software modules characterized by various software metrics and corresponding defect labels. These datasets have been extensively used in software engineering research and facilitate the evaluation of software defect prediction techniques under diverse project conditions.

The use of multiple datasets enhances the generalizability of the proposed framework and demonstrates its applicability to different software projects. The inclusion of the JM1 dataset further strengthens the evaluation framework due to its large number of software modules and extensive use in software defect prediction research. The diversity of the selected datasets enables the proposed EESDP framework to be applicable across software projects with varying sizes, complexities, and defect distributions.

The proposed framework is expected to support software quality assurance activities by facilitating the early identification of defect-prone software modules. Early defect prediction allows software organizations to allocate testing resources more efficiently, prioritize high-risk modules, reduce maintenance effort, and improve software reliability. By enabling proactive quality management, the framework can contribute to the development of dependable and maintainable software systems.

Compared with conventional single-classifier approaches, the EESDP framework provides a more comprehensive and systematic strategy for software defect prediction. The integration of preprocessing, feature ranking, and ensemble learning contributes to improved robustness, scalability, and adaptability. The framework is therefore suitable for practical software engineering environments where accurate defect prediction is essential for effective project management and software quality assurance.

Although the proposed framework offers several advantages, there remain opportunities for further enhancement. Future improvements may include the incorporation of Explainable Artificial Intelligence (XAI) techniques to improve

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 9, June 2026
Impact Factor: 8.2

model interpretability and assist software engineers in understanding the factors contributing to software defects. Additionally, deep learning architectures, optimization algorithms, hybrid ensemble techniques, and transfer learning approaches may be explored to further improve prediction performance.

Overall, the proposed EESDP framework provides a structured and scalable approach for software defect prediction by integrating software metrics analysis with ensemble machine learning techniques. The framework establishes a strong foundation for future experimental investigations and contributes to the advancement of intelligent software quality assurance methodologies.

VI. CONCLUSION AND FUTURE WORK

Software defect prediction has emerged as an important research area in software engineering due to its potential to improve software quality, reduce maintenance costs, and support efficient allocation of testing resources. Early identification of defect-prone software modules enables software organizations to focus quality assurance efforts on critical components and improve the reliability of software systems.

This paper presented an Enhanced Ensemble Software Defect Prediction (EESDP) framework that integrates software metrics, data preprocessing, feature ranking, and ensemble machine learning techniques for software defect prediction. The proposed framework combines the strengths of Random Forest and Gradient Boosting classifiers through a voting-based ensemble mechanism. Furthermore, the framework incorporates preprocessing and feature ranking strategies to improve data quality and enhance the effectiveness of the classification process.

The proposed EESDP framework was designed using benchmark NASA Metrics Data Program (MDP) datasets, including CM1, KC1, KC2, MC1, PC1, and JM1. The framework provides a systematic approach for identifying defect-prone software modules by leveraging software metrics and ensemble learning techniques. The integration of multiple classification strategies improves robustness and offers a practical solution for software quality assurance and maintenance planning.

The study highlights the significance of software metrics and ensemble learning in developing reliable software defect prediction systems. The proposed framework can assist software developers, testers, and project managers in making informed decisions regarding software quality management and risk assessment. By supporting early defect detection, the framework contributes to the development of dependable and maintainable software systems.

Future research can focus on the implementation and comprehensive experimental validation of the proposed framework using additional software defect datasets and advanced machine learning techniques. The integration of Explainable Artificial Intelligence (XAI) methods may improve the interpretability of defect prediction models, enabling software engineers to better understand the factors contributing to software defects. Furthermore, deep learning architectures, optimization algorithms, and hybrid ensemble techniques can be explored to enhance prediction performance. Future studies may also investigate cross-project software defect prediction, transfer learning approaches, and real-time defect prediction systems for large-scale software development environments.

In conclusion, the proposed EESDP framework provides a scalable and systematic foundation for software defect prediction and offers promising opportunities for improving software quality assurance practices through intelligent data-driven decision-making.

ACKNOWLEDGMENT

We express our heartfelt gratitude to Prof. Dr. P. S. Raghupathi for his constant encouragement, insightful guidance, and invaluable support throughout the research process. We also sincerely thank Dr. Indhu P. Nair, Principal, College of Engineering Kidangoor, Kottayam, for her continuous support, encouragement, and motivation in the successful completion of this research work.

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 9, June 2026
Impact Factor: 8.2

REFERENCES

[1] T. Menzies, J. Greenwald, and A. Frank, "Data mining static code attributes to learn defect predictors," IEEE Transactions on Software Engineering, vol. 33, no. 1, pp. 2–13, 2007.
[2] N. E. Fenton and M. Neil, "A critique of software defect prediction models," IEEE Transactions on Software Engineering, vol. 25, no. 5, pp. 675–689, 1999.
[3] T. J. Ostrand, E. J. Weyuker, and R. M. Bell, "Predicting the location and number of faults in large software systems," IEEE Transactions on Software Engineering, vol. 31, no. 4, pp. 340–355, 2005.
[4] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A systematic literature review on fault prediction performance in software engineering," IEEE Transactions on Software Engineering, vol. 38, no. 6, pp. 1276–1304, 2012.
[5] D. Radjenović, M. Heričko, R. Torkar, and A. Živkovič, "Software fault prediction metrics: A systematic literature review," Information and Software Technology, vol. 55, no. 8, pp. 1397–1418, 2013.
[6] M. A. Catal and B. Diri, "A systematic review of software fault prediction studies," Expert Systems with Applications, vol. 36, no. 4, pp. 7346–7354, 2009.
[7] T. M. Khoshgoftaar and N. Seliya, "Comparative assessment of software quality classification techniques," Empirical Software Engineering, vol. 9, no. 3, pp. 229–257, 2004.
[8] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th ed. Burlington, MA, USA: Morgan Kaufmann, 2016.
[9] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2012.
[10] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[11] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
[12] T. Dietterich, "Ensemble methods in machine learning," in Multiple Classifier Systems, Springer, 2000, pp. 1–15.
[13] E. Arisholm, L. Briand, and M. Fuglerud, "Data mining techniques for building fault-proneness models in telecom Java software," in Proceedings of ISSRE, 2007, pp. 215–224.
[14] Y. Kamei, E. Shihab, B. Adams, A. E. Hassan, A. Mockus, A. Sinha, and N. Ubayashi, "A large-scale empirical study of just-in-time quality assurance," IEEE Transactions on Software Engineering, vol. 39, no. 6, pp. 757–773, 2013.
[15] D. Bowes, T. Hall, and S. Beecham, "SLuRp: A software defect prediction dataset repository," Journal of Systems and Software, vol. 122, pp. 434–440, 2016.
[16] T. Menzies, A. Butcher, A. Marcus, T. Zimmermann, and D. Cok, "Local versus global lessons for defect prediction and effort estimation," IEEE Transactions on Software Engineering, vol. 39, no. 6, pp. 822–834, 2013.
[17] S. Lessmann, B. Baesens, C. Mues, and S. 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.
[18] M. Shepperd, D. Bowes, and T. Hall, "Researcher bias: The use of machine learning in software defect prediction," IEEE Transactions on Software Engineering, vol. 40, no. 6, pp. 603–616, 2014.
[19] R. Malhotra, "A systematic review of machine learning techniques for software fault prediction," Applied Soft Computing, vol. 27, pp. 504–518, 2015.
[20] S. Wang, T. Liu, and L. Tan, "Automatically learning semantic features for defect prediction," in Proceedings of the International Conference on Software Engineering, 2016, pp. 297–308.
[21] Rajeev P. R. and K. Aravinthan, "A Generic Approach for Software Metrics Based Software Defect Prediction," Indian Journal of Natural Sciences, vol. 15, no. 86, 2024.

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 9, June 2026
Impact Factor: 8.2

[22] Rajeev P. R. and K. Aravinthan, "A Novel Approach for Metrics-Based Software Defect Prediction Using Genetic Algorithm," Scientific Temper, vol. 15, no. 3, 2024.
[23] Rajeev P. R. and Sruthi R. Varrier, "A Deep Learning Enabled Improved Hybrid Genetic Based Rule Mining Framework for Software Defect Prediction," Accepted for Publication, 2026.
[24] A. Tosun, A. Bener, and B. Turhan, "Practical considerations in deploying machine learning techniques for software quality prediction," Information and Software Technology, vol. 52, no. 11, pp. 1165–1179, 2010.
[25] N. Nagappan and T. Ball, "Use of relative code churn measures to predict system defect density," in Proceedings of ICSE, 2005, pp. 284–292.

AUTHOR BIOGRAPHIES

Dr. Rajeev P. R. received the MCA degree from Mahatma Gandhi University, Kottayam, India, in 2014, the M.Phil. degree in Computer Science from Bharathidasan University in 2019, and the Ph.D. degree in Computer Science from Bharathidasan University in 2025. He is currently working as Assistant Professor in the Department of Computer Applications, College of Engineering Kidangoor, Kerala, India. He has more than six years of teaching experience in higher education institutions. His research interests include software engineering, software defect prediction, machine learning, deep learning, data analytics, and artificial intelligence. He has published research articles in national and international journals and conferences and serves as a reviewer for international conferences in the area of intelligent systems and software engineering.

Sruthi R. Varrier received the B.Tech. degree in Computer Science and Engineering from the University of Calicut, India, in 2018 and the M.Tech. degree in Computer Science and Information Systems from APJ Abdul Kalam Technological University, Kerala, India, in 2021. She is currently working as Assistant Professor in the Department of Electrical and Computer Engineering, College of Engineering Kidangoor, Kerala. She has industry experience as a Software Engineer and has also served as a Coding Educator and Mentor. Her research interests include machine learning, artificial intelligence, software engineering, data analytics, cybersecurity. She has authored and co-authored research publications in software defect prediction and applied artificial intelligence.