

# Chapter 10

## Software Defect Prediction Using Machine Learning Techniques

**G. Cauvery**


*St. Joseph's College of Arts and Science for Women, India*

**Dhina Suresh**

 <https://orcid.org/0000-0001-7504-8955>

*St. Joseph's College of Arts and Science for Women, India*

**G. Aswini**

 <https://orcid.org/0000-0002-3082-1956>

*St. Joseph's College of Arts and Science for Women, India*

**P. Jayanthi**

 <https://orcid.org/0009-0006-8239-0904>

*St. Joseph's College of Arts and Science for Women, India*

**K. Kalaiselvi**

*St. Joseph's College of Arts and Science for Women, India*

### ABSTRACT

*Software defect prediction gives development teams observable results while influencing business outcomes and development flaws. Developers can uncover flaws and plan test activities by anticipating problematic code sections. Early identification depends on the percentage of classifications that make the right prediction. Additionally, software-defective data sets are supported and partially acknowledged because of their vast size. The confusion, precision, recall, identification accuracy, etc., are assessed and compared with the existing schemes in a systematic research analysis. Previous research has employed the weak simulation tool for software analysis, but this study proposes building three machine learning models using linear regression, KNN classifier, and random forest (RF). According to the analytical investigation, the suggested approach will offer more beneficial options for predicting device failures. Moreover, software-defected data sets are supported and at least partially recognized due to their enormous dimension.*

DOI: 10.4018/979-8-3693-1301-5.ch010

## **1. INTRODUCTION**

Creating a software system nowadays requires careful planning, analysis, implementation, testing, integration, and maintenance (Viswakarma, et al., 2014). The task of a software Engineer is to design a system within a set deadline and budget, which is accomplished during the planning stage (Alajmi & Khan, 2013). We may encounter a few flaws during the development process, such as bad design, faulty logic, improper data processing, etc.; these flaws result in errors that force us to redo the job, driving up the cost of development and maintenance (Chitra, et al., 2018). All of these are to blame for the decline in consumer satisfaction (Garg, et al., 2017). According to this viewpoint, faults are categorized to their severity, and corrective and proactive measures are implemented following the severity determined (Chopra, et al., 2022). Humans have increasingly centered their attention on software-based systems over the past ten years, with software quality being seen as the most important factor in user functionality (Hung & Chakrabarti, 2022).

Subpar results for both commercial and personal apps can be attributed to a lack of software quality, despite the widespread development of application software (Jain, et al., 2023). Defect prediction designs are widely used in industries, and the models they produce help with tasks like fault prediction, effort estimation for software reliability testing, hazard analysis, and more (Jayalakshmi & Ramesh, 2020). A supervised machine-learning forecasting algorithm is fed the predefined training data set (Kayalvizhi & Ramesh, 2020). The algorithm then produces rules based on what it has learned from the training dataset in order to predict the class label for a new data set (Tiwari, et al., 2018). Mathematical approaches are utilised during the learning phases to develop and enhance the prediction function (Khan, 2016). This technique employs training data with a known attribute input value and known output value (Khan, 2021). The quality of the expected ML algorithm is compared to the widely known result (Oak, et al., 2019). Training data is used to perform this process over and over again until the highest possible prediction accuracy is reached or the maximum number of loops is reached (Prasad, et al., 2013).

In unsupervised learning methods, the actual value of the class label output is not known beforehand. Early on in the process, faulty systems, such individual units or entire classes, can be identified with the use of defect prediction modelling (Rajeyyagari, et al., 2022). This can be done by labelling the modules as either reliable or prone to errors. Different methods, including support vector classifiers (SVC), random forests, naive Bayes, decision trees (DT), and neural networks, are used to select the classification module (NN). The modules most prone to defects are given higher priority throughout the progress testing phases, whereas the modules least prone to defects are examined as resources permit (Sandeep, et al., 2022).

The classifier approach establishes and examines the feature of classification known as the relationship between the attributes and the training dataset class label through formulas for categorizing the targets. Future dataset class labels must be defined using those guidelines as well (Vanitha, et al., 2019). As a result, the unclassified datasets can be categorized using a classifier and classification patterns (Prasad & Chakrabarti, 2014). Due to the widespread use of software, defining software problems, locating the defect, and recognizing it requires researchers to perform repetitive tasks. The primary objective of separating the software dataset into a faulty and non-defective dataset is to use it as a model for bug prediction.

This strategy entails providing the Classifier with the input programme dataset only after the user has determined the correct class values (Shah, et al., 2020). Prior to this strategy, tremendous progress was made using requirement- and design-based metric approaches. However, developing algorithms and improving the accuracy of predictions remain difficult jobs (Kumar, et al., 2020). However, the results

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/software-defect-prediction-using-machine-learning-techniques/330405](http://www.igi-global.com/chapter/software-defect-prediction-using-machine-learning-techniques/330405)

## Related Content

---

### Multimedia, Scientific Information, and Visualization for Information Systems and Metrics

Francisco V. Cipolla-Ficarra (2018). *Technology-Enhanced Human Interaction in Modern Society* (pp. 56-78).

[www.irma-international.org/chapter/multimedia-scientific-information-and-visualization-for-information-systems-and-metrics/189837](http://www.irma-international.org/chapter/multimedia-scientific-information-and-visualization-for-information-systems-and-metrics/189837)

### Technique and Behaviors, Trendy, Useful, Correct: From Unified Procedures to Unified Thinking?

(2024). *Considerations on Cyber Behavior and Mass Technology in Modern Society* (pp. 123-147).

[www.irma-international.org/chapter/technique-and-behaviors-trendy-useful-correct/338253](http://www.irma-international.org/chapter/technique-and-behaviors-trendy-useful-correct/338253)

### Free Emails in Bad Portals

Francisco V. Cipolla-Ficarra, Alejandra Quirogaand Valeria M. Ficarra (2018). *Technology-Enhanced Human Interaction in Modern Society* (pp. 234-262).

[www.irma-international.org/chapter/free-emails-in-bad-portals/189846](http://www.irma-international.org/chapter/free-emails-in-bad-portals/189846)

### Artificial Neural Networks

Steven Walczak (2019). *Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction* (pp. 40-53).

[www.irma-international.org/chapter/artificial-neural-networks/213116](http://www.irma-international.org/chapter/artificial-neural-networks/213116)

### Ecological Perspectives Surrounding the Design of Self-Determination-Enhanced Problem-Based Learning as a Formative Intervention for Students with Disabilities in Inclusive Settings

Soohnwa Seok, Boaventura DaCostaand Woo Kim (2016). *Human-Computer Interaction: Concepts, Methodologies, Tools, and Applications* (pp. 330-348).

[www.irma-international.org/chapter/ecological-perspectives-surrounding-the-design-of-self-determination-enhanced-problem-based-learning-as-a-formative-intervention-for-students-with-disabilities-in-inclusive-settings/139041](http://www.irma-international.org/chapter/ecological-perspectives-surrounding-the-design-of-self-determination-enhanced-problem-based-learning-as-a-formative-intervention-for-students-with-disabilities-in-inclusive-settings/139041)