SOFTWARE RELIABILITY PREDICTION USING MACHINE LEARNING TECHNIQUES

## Mrs.T.SUNEETHA [1]K.SAIKIRAN [2]G.SAIPRABHAS [3]V.NIKHIL

Assistant Professor Student/Research Scholars

Department of Cyber Security Malla Reddy University,Hyderabad

Maisammaguda, Dulapally,Hyderabad,500100, Telangana, India

#

# 2111cs040084@mallareddyuniversity.ac.in

# 2111cs040090@mallareddyuniversity.ac.in

# 2111cs040062@mallareddyuniversity.ac.in

**ABSTRACT**

**Software reliability is a critical component of software quality, serving as a fundamental criterion for evaluating the overall performance and user satisfaction of software products. The software industry faces numerous challenges in delivering highly reliable systems, necessitating robust methodologies for reliability assessment and improvement. In this study, we employ advanced ensemble methods and machine learning techniques to predict software reliability, systematically evaluating these approaches against key performance criteria.In our proposed work, we explore all the software jobvarious ensemble methods—including bagging, boosting, and stacking—across different machine learning algorithms.**

**I.INTRODUCTION**

Today, more and more of the world around us is computerized. Along with this, more and more sensitive operations are computer controlled, for example medical monitoring, economical transactions etc. Concurrently we increase our confidence in that the software responds correctly. Accordingly, our expectations on software reliability are assumed to be obvious and are therefore one of the most important software quality attributes.To assess a certain reliability level we have to determine software reliability. Software usage varies depending on several factors like type of software, type of environment and so forth. Still there is a need of capturing a usage that, as good as possible, represents the main purpose of the software. Software reliability modelling based on test data is done to estimatewhetherthe current reliability level meets the requirements for the product software reliability modelling also provides possibility to predict the reliability of the modules in a software. In this proposed work, the model has been used for predicting and estimating the number of errors.Software reliability modelling based on test data is done whether the current reliability level meets the requirements for reliability modelling also provides possibility to predict the reliability of the modules in a software. In this proposed work, the model has been used for predicting and estimating the number of errors in the software. Also, classification is also done in this work to classify the error to a desired class of output. In this proposed work, implementation is done through Machine Learning Techniques and Ensembling Methods for predicting software reliability. Machine Learning(ML) techniques have proved to

be successful in predicting better results that statistical methods and can be used for prediction and classification of software failures more accurately and precisely. ML is an approach which is focused on learning automatically and allows computers to evolve and predict and classify the system behavior based on past and the present failure data. Thus, it is quite natural to know that which method tends to work well for a given failure dataset and up to what equantitatively .In this proposed work, we present an ensemble of various ML techniques such as Bagging, Boosting & Stacking for predicting and classifying Software reliability based on two industrial datasets. Thereafter, we check about the accuracy and performances of ML based models in predicting and classifying Software reliability when applied to past failure week data.In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent algorithms alone. Machine Learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

## II.PROBLEM STATEMENT

Software reliability is a critical factor in software development, as it directly impacts user satisfaction, maintenance costs, and overall system performance. Traditional methods of measuring software reliability often rely on historical data and statistical techniques, which may not effectively capture the complexities of modern software systems. With the rise of machine learning (ML), there is an opportunity to enhance reliability predictions by leveraging large datasets and advanced algorithms.The primary objective of this project is to develop a machine learning model that predicts software reliability based on various metrics collected during the software development lifecycle. The model should be able to forecast the likelihood of software failures, helping developers and project managers make informed decisions regarding testing, deployment, and maintenance.Identify the most significant features that influence software reliability. This may involve using techniques such as correlation analysis, feature importance scoring, and dimensionality reduction.Utilize various machine learning algorithms (e.g., decision trees, random forests, support vector machines, neural networks) to build a predictive model. The model should be trained on a portion of the data and validated on a separate set to evaluate its performance.By applying machine learning techniques to predict software reliability, this project aims to enhance the overall quality of software systems, reduce maintenance costs, and improve user satisfaction through proactive management of software defects.Develop methods to interpret the model's predictions, ensuring that stakeholders can understand the factors contributing to the reliability predictions.Create a framework for integrating the predictive model into existing software development workflows, providing real-time reliability assessments during the development process.

## METHODOLOGY

Dataset is a collection of data.In our project we are using two Datasets for software reliability prediction.For prediction of time of failure, dataset used: Musa, J.D: Software reliability data, IEEE Computer Society Repository(1979). Failure data during system testing phase of various projects collected at Bell Telephone Laboratories, Cyber security and Information Systems Information Analysis Centre(CSIAC) by John D. Musa are considered.For prediction of time of failure, dataset used: Musa, J.D: Software reliability data, IEEE Computer Society Repository(1979). Failure data during system testing phase of various projects collected at Bell Telephone Laboratories, Cyber security and Information Systems Information Analysis Centre(CSIAC) by John D. Musa are considered.It is a single feature dataset representing the meantime between failures in chronological order. This methodology outlines a systematic approach to developing a machine learning model for software reliability prediction. By leveraging data-driven insights, organizations can enhance their software quality, minimize defects, and ultimately deliver more reliable products to users.This methodology outlines a systematic approach to predicting software reliability using machine learning techniques. First, the problem is defined by identifying objectives and key stakeholders to understand their needs. Data collection follows, sourcing quantitative and qualitative information from version control systems, issue tracking tools, and code analysis. The data is then preprocessed, which involves cleaning, normalizing, and encoding categorical variables to prepare for analysis.

## REFERENCES

1. **A Survey of Software Reliability Prediction ;**Authors: M. Lyu, Y. S. Lee, and others.Source: ACM Computing SurveysSummary: This paper surveys various software reliability prediction models, including machine learning techniques applied in reliability prediction.

**[2]A Comparative Study of Machine Learning Algorithms for Software Reliability Prediction:**Source: IEEE AccessSummary: This paper compares the performance of several machine learning algorithms, including decision trees, support vector machines, and neural networks, for predicting software reliability.

[3]**Predicting Software Reliability: A Machine Learning Approach**Source: Information and Software Technology JournalSummary: The paper explores the use of different machine learning models for predicting software reliability based on historical defect data.

[4]Software **Reliability Engineering: More Than Just a Number** Author: John D. Musa, Anthony Iannino, Kazuhiko OkumotoSummary: Focuses on software reliability models and includes a discussion of the transition to more modern techniques, including machine learning approaches.

[5]Pattern **Recognition and Machine Learning;**Author: Christopher M. Bishop Summary: This book covers essential machine learning methods and techniques like classification, regression, and clustering that can be applied to software reliability prediction.

1. **FUTURE SCOPE**

The future of software reliability prediction using machine learning techniques is poised for significant advancements. As machine learning models become more sophisticated, they will be able to predict failures with greater accuracy, adapt to real-time changes in software systems, and offer deeper insights into the root causes of reliability issues. These advances will ultimately lead to more reliable, efficient, and maintainable software, helping organizations mitigate risks, improve user experiences, and optimize development processes. With the rapid pace of progress in AI and machine learning, the potential for predictive reliability models will continue to expand, opening new avenues for improving the robustness and resilience of software applications across industries.

## ARCHITECTURAL DESIGN

The architectural design of a Software Reliability Prediction system using Machine Learning techniques is a multi-layered approach that involves KNN,Jupyter, data collection, preprocessing, model training, prediction generation, continuous learning, and seamless integration with existing development workflows. The goal is to build a system that can dynamically predict and improve software reliability by leveraging historical data, real-time metrics, and advanced machine learning algorithms. This architecture not only improves the ability to predict software failures but also empowers development teams to make data-driven decisions to enhance software quality, reduce maintenance costs, and ensure continuous improvement in software reliability over time.

1. **SYSTEM ANALYSIS**
	1. Existing System

The existing system for software reliability prediction, which rely heavily on statistical models and expert judgment, have several limitations that can impact their effectiveness in modern, complex software environments.**Poisson Process Models** (e.g., Jelinski-Moranda, Goel-Okumoto models)**Non-Homogeneous Poisson Process (NHPP)** models**Markov ChainsBayesian Network Models.**NASA has developed several reliability models, primarily based on statistical techniques and historical data.**FMEA** are other traditional approaches used to identify reliability risks based on expert judgment.The **existing systems** for software reliability prediction have been predominantly based on **traditional statistical models**, **expert-based systems**, and **heuristic approaches**.Existing systems for software reliability prediction, though foundational and valuable in specific contexts, have several limitations, especially when applied to modern, complex software development practices.

* 1. Proposed System

The **proposed system**, which incorporates machine learning techniques, offers a more dynamic, scalable, and accurate approach to software reliability prediction. By continuously adapting to real-time data and providing deeper insights into software performance, this ML-based system can significantly improve software quality and reduce the likelihood of failures, making it a vital tool in modern software engineering practices.The proposed system aims to incorporate **Machine Learning (ML)** techniques into software reliability prediction, which will overcome the limitations of existing systems. ML-based approaches will provide more accurate, dynamic, and scalable predictions, taking advantage of large datasets and real-time system performance metrics.The proposed system will integrate various ML models with data collection, processing, and prediction workflows to forecast software reliability more effectively.

## RESULTS





 **XI. SOFTWARE REQUIREMENTS**

**Jupyter:**

Jupyter is a non-profit, open-source project, born out of the [IPython Project](https://ipython.org/) in 2014 as it evolved to support interactive data science and scientific computing across all programming languages. Jupyter will always be 100% open-source software, free for all to use and released under the liberal terms of the [modified BSD license](https://opensource.org/licenses/BSD-3-Clause).Jupyter's operating philosophy is to support interactive [data science](https://en.wikipedia.org/wiki/Data_science) and [scientific computing](https://en.wikipedia.org/wiki/Computational_science) across all [programming languages](https://en.wikipedia.org/wiki/Programming_language) via the development of [open-source software](https://en.wikipedia.org/wiki/Open-source_software). According to the Jupyter website, "Jupyter will always be 100% open-source software, free for all to use and released under the liberal terms of the modified BSD license".Jupyter Notebook provides a browser-based [REPL](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop) built upon a [source](https://en.wikipedia.org/wiki/Open-source_software) libraries :IPython,OMQ,Tornodo(web server),Mathjax

**Features of jupyter:**

The Jupyter Notebook is a very handy coding tool for data scientists. It allows us to visualize data in the form of text and images while we’re moving forward with the processing and analysis of our datasets.Some of the remarkable features of jupyter are:

**1.Run Commands**

To run commands in cells, we can simply prefix an exclamation mark before the command. One common question associated with running commands is how we interact with prompts, such as installation confirmation (yes or no), during some execution. The trick is to append the yes flag (-y) to the command.

**2.Check Active Variables**

When we work with lots of data, we may have created many intermediate variables in our workspace. To get to know the current pool of these variables, we can use the magic method %whoor %whos.

**3.API Lookups**

We don’t always remember the functions or attributes that we want to use. However, we do have some impressions about them, after all, we may have used them from time to time before. In this case, we can list all related methods that may remind us.

**4.Change the Default Output Mode**

By default, each cell only prints the last expression after executing the entire cell.However, chances are that we want to display more items, such as all the expressions in the cell. In this case, we can change the default setting that allows the output of multiple items.

 **Operating System**

 Windows/Linux operating system

 **Hardware Requirements**

Minimum of 1GB Hard Disk

Minimum of 4GB RAM

 X.  **X.CONCLUSION**

Different machine learning prediction and classification algorithms are used in ensemble learning for software reliability prediction. As we have seen that in the bagging approach of the classification the output is classified in different class and then voting method is used to get the final output. In prediction problem we are getting very less error rate for prediction. In boosting model, we have taken weak machine learning model and then by using AdaBoost algorithm the accuracy of the model is improved. In stacking we are combining different base learner algorithm and then predicting the dataset and finally we are using combiner algorithm to predict the output. All the output value is stored in xlsx file and tableau software is used to visualize the data or the output that we got from the Ensemble method for software reliability prediction.In bagging method of classification, we can see that for the decision tree gives a highest performance among all other machine learning model. In boosting method, KNN gives high Accuracy. For stacking KNN model gives highest accuracy among all.