IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 5, May 2023

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205065 |

Software Bug Prediction by Tuning XG Boost Algorithm Using Machine Learning

Dr. M. Srinivasan, Mrs. T. Jansi Rani, Ms.R.Abisha, Ms.K.Minithasri, Ms.V.Rajalakshmi

Professor, Department of Information Technology, PSV College of Engineering and Technology, Tamil Nadu, India

Assistant Professor, Department of Information Technology, PSV College of Engineering and Technology, Tamil Nadu, India

Department of Information Technology, PSV College of Engineering and Technology, Tamil Nadu, India

Department of Information Technology, PSV College of Engineering and Technology, Tamil Nadu, India

Department of Information Technology, PSV College of Engineering and Technology, Tamil Nadu, India

ABSTRACT- As the number of Internet users increases, so does the amount of data available on the web. Virtually anything that requires human effort or human presence can be replaced by Software. When developing an application, it follows the Software Development Life Cycle (SDLC). In the early stages of development, it is a mandatory task to take care of the system or bugs to avoid wasting time and effort during the initial development phase to avoid any runtime crisis. In this project we used machine learning models – Logistic Regression, Decision Tree, Random Forest. The results achieved were compared with existing models and our model outperformed them on all datasets. The project believes that this research will contribute to correct error detection using a machine learning approach.

KEYWORDS : machine learning, dataset, supervised learning, random forest and decision tree.

I.INTRODUCTION

The existence of software bugs dramatically affects software reliability, quality, and maintenance costs. Achieving bug-free software is also hard work, even with carefully applied software, because there are usually hidden bugs. Anticipating software bugs is a fundamental activity in software development. This is because predicting faulty modules before software deployment achieves user satisfaction and improves overall software performance. In addition, early prediction of a software error improves the adaptability of software to different environments and increases resource utilization. Various techniques have been proposed to solve the software bug prediction (SBP) problem. The most well-known techniques are machine learning (ML) techniques. Section 2 presents a discussion of related work in SBP. An overview of selected ML algorithms is given. Describes data sets and evaluation methodology.

II.OBJECTIVE

• Developing a robust error prediction model is a challenging task and many techniques have been proposed.
• This project presents a software error prediction model based on machine learning (ML) with random forest algorithms.
• Anticipate Software Error Prediction in software development and maintenance processes that relate to overall success of the software.
• The goal is to analyze and assess three supervised machine learning algorithms, which are the Random Forest algorithm and the Decision Tree (DT).

III.LITERATURE REVIEW

A lot of effort and resources are spent on fixing bugs. Sometimes these errors or defects can be seen in the form of vulnerability and hackers will use it as a clue to abuse your website or application and sometimes you will compromise important information or money. Thus, it reveals the problem facing the software industry and the

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205065 |

importance of anticipating software defects at the stage of its development.[1] Basically, no human-developed application is an automated process, so having a defect is a common or natural thing. Nevertheless, software development companies focus on early detection of defects using several checks, testing procedures. To address this issue, we reviewed various machine learning-based approaches.[2] Focused on predicting web application vulnerabilities using machine learning. Input validation and hygiene attributes are generated in this document. Calculates the static backcut for each sink. Program analysis is based on the control flow graph, control dependency graph, and system dependency graph of the web program.[3] A model for an object-oriented software fault prediction system (SBPS). The study combined similar types of defect datasets available in the Promise Software Engineering Repository. The study evaluated the proposed model using performance (accuracy) measures. Finally, the results of the study showed that the average accuracy of the proposed model is 76.27%.[4] An application using a genetic algorithm to predict susceptibility to failure. The application obtains its values, such as object-oriented metrics and count metric values, from an open source software project. The genetic algorithm uses application values as inputs to generate rules that are used to categorize software modules into defective and non-defective modules. Finally, visualize the outputs using the Genetic Algorithm applet.[5]

IV.EXISTING SYSTEM

- Software error estimation is done to identify software modules that may have some errors.
- The cost of fixing bugs is high and it is recommended to start predicting bugs in the later stages of development rather than testing.
- Less efficient methods and techniques, such as machine learning, are available for building predictive models.
- This method is not widely used today as it provides accurate results and analysis.
- Developing predictive models poses some challenges.
- Solving the algorithm takes a lot of time, the algorithm is difficult to understand.

V.PROPOSED SYSTEM

- The recommended system will predict bugs for certain software.
- After clearing the data, the sklearn library will be used to split the data into trains and tests.
- Machine learning prediction algorithms have very good predictions that need to be based on data.
- The aim is to go beyond knowing what has happened to provide the best possible assessment of what will happen in the future.
- After collecting information from various sources. Data must be processed before training the model.
- Data processing can be carried out in several stages, starting with reading the archive and continuing with the data cleaning process. In data cleaning, there are some inconsistencies in the dataset that are not considered errors.
- So we need to remove the unwanted features and there are some missing values in the data, we need to delete the missing values or write the unwanted values in order to better accept the truth.
- After initial data, use Random Forest to train the model to the training set.
- We consider software wrong guess for wrong guess.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205065 |

VI. ARCHITECTURE DESIGN

A. BLOCK DIAGRAM

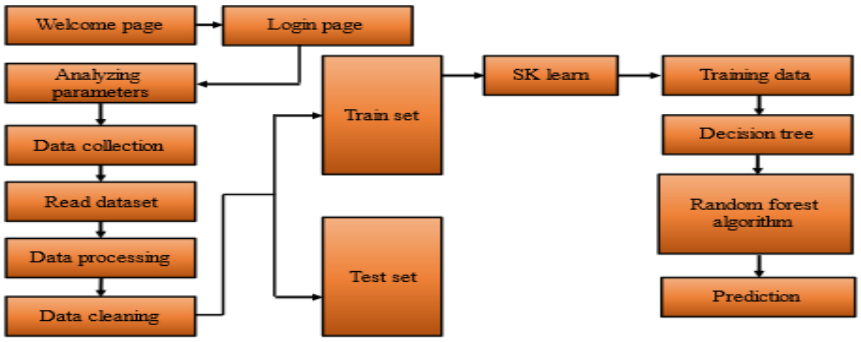

Fig no: 1 proposed for architecture design

VII. MODULES OF DESCRIPTION

Six modules follow
- Welcome module
- Data collection module
- Data processing module
- Learning module SK
- Random Forest module
- Prediction module

A. MODULE DESCRIPTION

Welcome module:

On welcome, when user opens the app, a welcome will be displayed on clicking next and a login page will be displayed. On this login page, the user should enter the correct username and password to enter the next page. If the username and password are incorrect, the page will remain permanently displayed.

Data collection module:

In this module, the software error data will be collected in a CSV file. This collected data will be visualized and checked using python whether the software errors match correctly or not.

Data processing module:

The collected data will not be accurate, so we will read the collected data and the data will be pre-processed, but we will remove the external points and errors of the analyzed data accurate errors in the software.

SK learn module:

In this module, the data train set and the test set will be implemented using the Sk Learn module. A random forest is a commonly used machine learning algorithm that combines the output of multiple decision trees to produce a single result. Its ease of use and flexibility have fueled adoption as solves both classification and regression problems.

Random forest module:

In this module data train set and test set will be implemented with Sk learn module. Random forest is a commonly-used machine learning algorithm, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205065 |

Prediction module:

In the prediction module, the user will know whether the weather is predicted in the software or not. By predicting bugs in software, we can keep software very secure. So anticipating bugs in software is very important to keep the application secure.

VIII.SYSTEM SOFTWARE

1.RANDOM FOREST

Random Forest is a machine learning technique used to solved regression and classification problems. It uses learning techniques, a technique that combines multiple classifications to provide solutions to complex problems. The random forest algorithm as many decision trees. Bagging is a set of meta algorithms improve the accuracy of machine learning algorithms. Makes predictions by averaging of various trees. Increasing the number of trees and accuracy of the results. It reduces dataset overfitting and improves accuracy. It generates predictions without much configuration in the package.

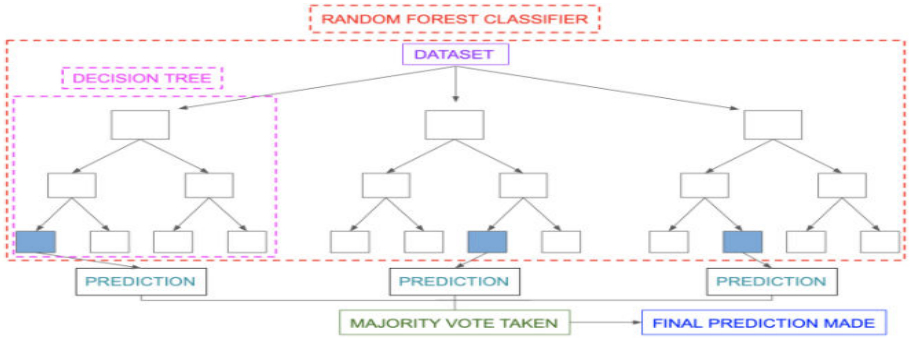

Fig no : 2 Random Forest Algorithm

2.Data collection

The data collection and evaluation process about defined and functioning variables leads to voluntary answering research questions, evaluating opinions and evaluating results. The literature portion of this work is available for all courses, including physical and social sciences, humanities, economics, and more. While the approach differs between disciplines, the focus on authenticity and fair writing remains the same. Whatever the study or data interpretation preference , accurate data collection is essential to maintain integrity. Both the selection appropriate data collection newly developed and clear instructions explaining correct used can reduce the possibility of error.

3.Data Processing

Data processing is the process of collecting raw data and transforming it into usable data. It is often one step ahead of the team of data scientists and data engineers in the enterprise. Data processing is essential for organizations to develop better business strategies and increase their competitive advantage. Data can be translated into readable formats such as charts, graphs, and documents, which can be understood and used by employees across the organization.Each step is done in a sequence, but the whole process is repeated in a loop. The output of the first processing cycle can be stored as shown in the figure below and used as input for the next cycle.

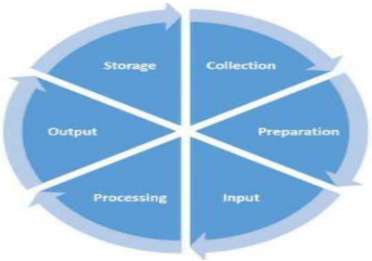

Fig no: 3 data processing cycle

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205065 |

4. SK LEARN

Scikit-learn (Sklearn) is the most useful and powerful machine learning library in Python. It provides many useful tools for machine learning and statistical models such as classification, regression, clustering and dimensionality reduction by interacting in Python. The library is primarily written in Python and is built on NumPy, SciPy, and Matplotlib. It was originally called scikit. It was learned and originally developed by David Cornopean in 2007 as part of the Google Summer of Code project. Then, in 2010, Fabian Pedrosa, Gael Viroqua, Alexandre Gram fort and Vincent Michel from FIRCA French Institute for Computer Science and Automation took the project to another dimension and published it publicly for the first time.

5. DECISION TREE

Decision tree learning uses the divide-and-conquer strategy, seeking the desire to determine the best distribution of points in the tree. This classification process is then repeated from top to bottom until all or most of the information is classified in a single tag. Although all data points are classified as homogeneous sets due to the complexity of the decision tree. For example, smaller trees are easier to get clean leaves. The content of the data in a class. However, as the tree grows, this purity becomes increasingly difficult to maintain, often resulting in too little information to fit into a particular tree.
When this happens, it's called data fragmentation and often leads to overfitting. Therefore, decision trees prefer smaller trees based on the stinginess principle of Occam's razor; i.e. "no spaces should be added unless necessary". In other words, decision trees add complexity only when necessary because simple explanations are often best.

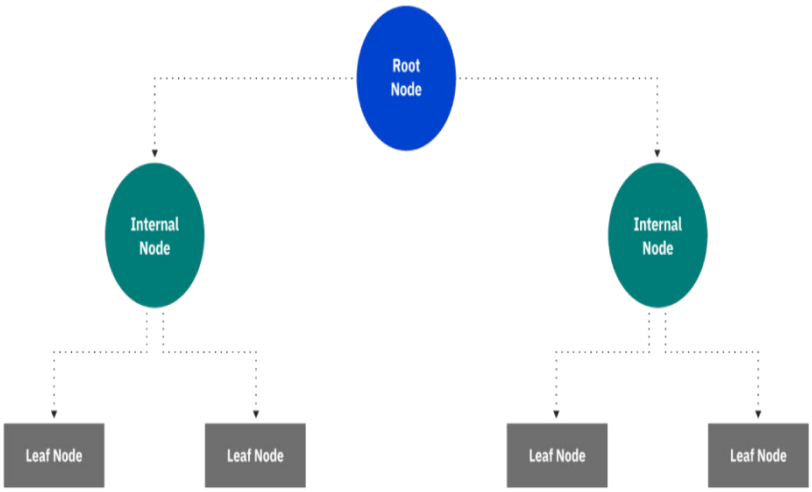

Fig no : 4 decision tree

6. DATA CLEANING

Data processing is a essential part of machine learning. It plays an important role in the creation of the structure. This is definitely not the coolest thing about machine learning, and there are no secrets to problem solving either. The success or failure of the project depends on the proper cleaning of the data. Professional data scientists often spend a lot of time on this step because they believe that "better data is better algorithms".The model sees and learns from this data to predict outcomes or make the right decisions. Training data is usually collected from various sources, then preprocessed and organized to ensure the performance of the model. The type of training data often determines the model's ability, i.e.A model is said to be overfitting if it is more involved in the training data than the test data. This information represents approximately 20-25% of the total information available for the project.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205065 |

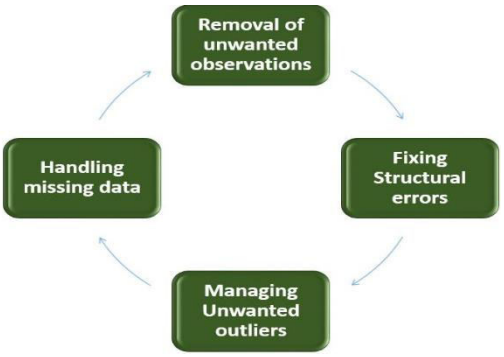

Fig no: 5data cleaning

7. TRAIN AND TEST SET

This is the real data set that the model trains from, i.e. the model sees and learns from this data to predict the outcome or make the right decisions. Most of the training data is collected from several sources and then pre-processed and organized to provide the correct performance of the model. The type of training data greatly determines the ability of the model to generalize, i.e. the better the quality and diversity of the training data, the better the performance of the model. This data constitutes more than 60% of the total data available for the project.

Test set

A test set is usually a well-organized data set containing all kinds of data for scenarios that the model would likely face in real-world use. A combination of validation and test suite is often used as a test suite, which is not considered a best practice. If the accuracy of the model on the training data is greater than on the test data, then the model is said to be overfitted. This data constitutes approximately 20-25% of the total data available for the project.

VIII. RESULT

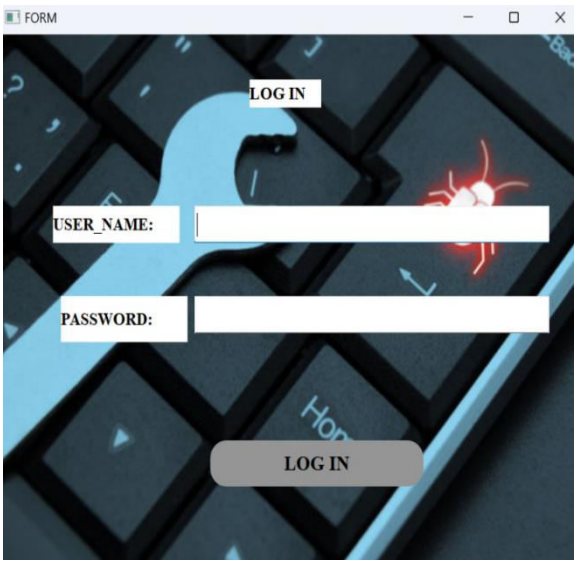

Fig no 6:Log in page

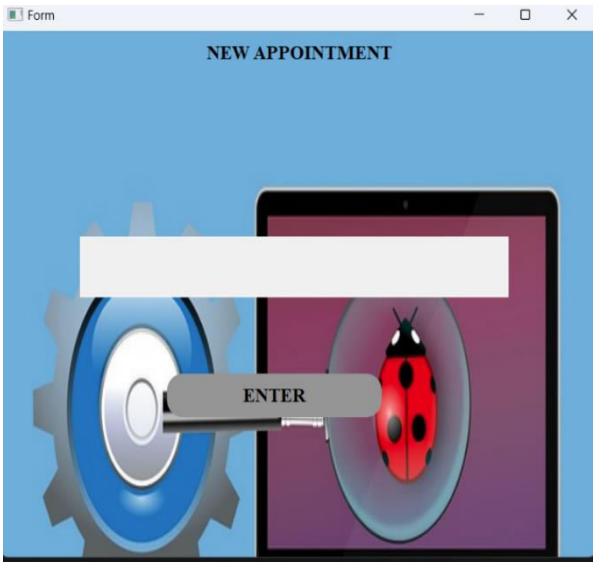

Fig no 7: Prediction

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205065 |

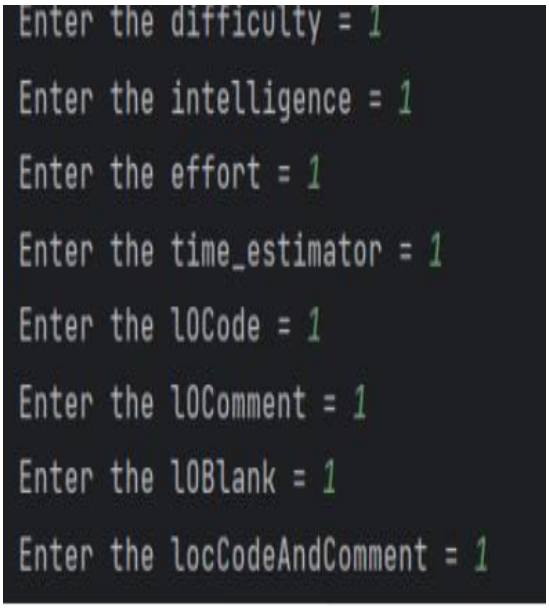

Fig no 8: Enter values

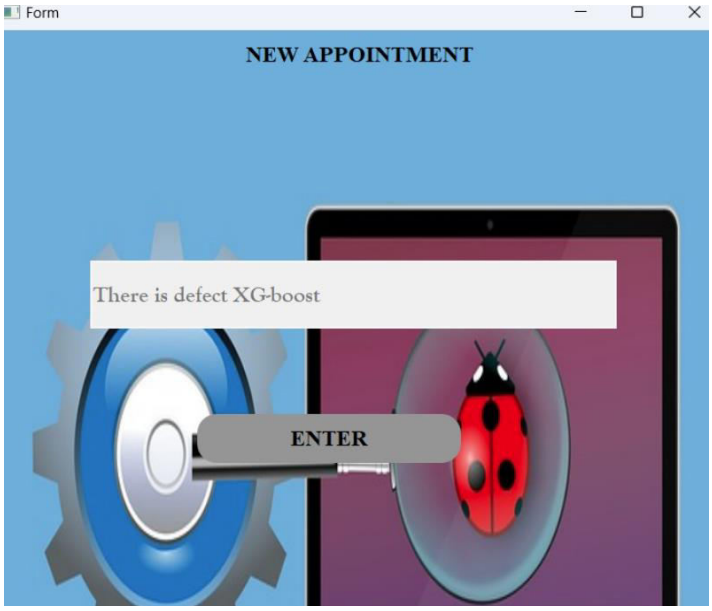

Fig no 9: Output

XI.CONCLUSION

Software bug prediction is a technique where a predictive model is built to predict future software bugs based on historical data. Different approaches have been proposed using different datasets, different metrics and different performance measures. This project evaluated the use of machine learning algorithms in the software bug prediction problem. Machine learning techniques such as DT and Random Forest algorithm were used. The results show that ML techniques are effective approaches for predicting future software errors. The comparison results showed that the DT classifier has the best results over the others.

REFERENCES

[1] Rashid, Mamoon, Lovepreet Kaur "Finding Bugs in Android Application using Genetic Algorithm and Apriori Algorithm." Indian Journal of Science and Technology (2019).

[2] Shruthi Puranika, Pranav Deshpandea, K Chandrasekarana "A Novel Machine Learning Approach for Bug Prediction" ScienceDirect, 2020.

[3] Ali Ouni, Marwa Daagi, Marouane Kessentini, Salah Bouktif, Mohamed Mohsen Gammoudi. "A Machine Learning-Based Approach to Detect Web Service Design Defects" IEEE, 2021.

[4] M. M. Rosli, N. H. I. Teo, N. S. M. Yusop and N. S. Moham, "The Design of a Software Fault Prone Application Using Evolutionary Algorithm," IEEE Conference on Open Systems, 2019.

[5] T. Gyimothy, R. Ferenc and I. Siket, "Empirical Validation of ObjectOriented Metrics on Open-Source Software for Fault Prediction," 2018.

[6] D'Ambros, Marco, Michele Lanza, and Romain Robbes. "An extensive comparison of bug prediction approaches." Mining Software Repositories (MSR), 2019 7th IEEE Working Conference on. IEEE, 2019.

[7] T. Gyimothy, R. Ferenc and I. Siket, "Empirical Validation of ObjectOriented Metrics on Open Source Software for Fault Prediction," IEEE Transactions On Software Engineering, 2019.

[8] Singh, Praman Deep, and Anuradha Chug. "Software defect prediction analysis using machine learning algorithms." 7th International Conference on Cloud Computing, Data Science & EngineeringConfluence, IEEE, 2020.

[9] M. C. Prasad, L. Florence and A. Arya, "A Study on Software Metrics based Software Defect Prediction using Data Mining and Machine Learning Techniques," International Journal of Database Theory and Application, pp. 179-190, 2018.

[10] Okutan, Ahmet, and Olcay Taner Yıldız. "Software defect prediction using Bayesian networks." Empirical Software Engineering 19.1 (2019): 154-181.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205065 |

[11] Bavisi, Shrey, Jash Mehta, and Lynette Lopes. "A Comparative Study of Different Data Mining Algorithms." International Journal of Current Engineering and Technology 4.5 (2019).
[12] Y. Singh, A. Kaur and R. Malhotra, "Empirical validation of objectoriented metrics for predicting fault proneness models," Software Qual J, p. 3–35, 2020.
[13] Malhotra, Ruchika, and Yogesh Singh. "On the applicability of machine learning techniques for object oriented software fault prediction." Software Engineering: An International Journal 1.1 (2011): 24-37, 2019.
[14] A.TosunMisirli, A. se Ba¸ S.Bener,"A Mapping Study on Bayesian Networks for Software Quality Prediction", Proceedings of the 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, (2018).
[15] T. Angel Thankachan1, K. Raimond2, "A Survey on Classification and Rule Extraction Techniques for Data mining",IOSR Journal of Computer Engineering ,vol. 8, no. 5,(2013), pp. 75-78.
[16] T. Minohara and Y. Tohma, "Parameter estimation of hyper-geometric distribution software reliability growth model by genetic algorithms", in Proceedings of the 6th International Symposium on Software Reliability Engineering, pp. 324–329, 2018.
[17] Olsen, David L. and Delen, " Advanced Data Mining Techniques ", Springer, 1st edition, page 138, ISBN 3-540-76016-1, Feb 2008.
[18] L. H. Crow, "Reliability for complex repairable systems," Reliability and Biometry, SIAM, pp. 379–410, 1974, 2018

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.423

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details