SOFTWARE DEFECT PREDICTION TO IMPROVE SOFTWARE QUALITY USING MACHINE LEARNING APPROACH

Dr.Katta Subba Rao¹, Dr.Ch.Madhu Babu², Govindu Surla³, J.Nageswara Rao⁴,
¹Professor, ²Professor & HOD,
¹,²Department of Computer Science & Engineering,
²B V Raju Institute of Technology, Narsapur, Medak (District), Telangana. Pin: 502313,
E mail Id: subbarao.k@bvrit.ac.in, madhubabu.ch@bvrit.ac.in,
³Assistant Professor, Department of Computer Science and Engineering,
³Lakireddy Bali Reddy College of Engineering, AP, India, 521230,
⁴Sr. Assistant Professor, Department of Computer Science and Engineering,
⁴Lakireddy Bali Reddy College of Engineering, AP, India 521230.

ABSTRACT

Software Engineering is most commonly used to build application-based programming, preparation, and deployment. For any technology organization, it is essential that even the most inexperienced software users must produce high-quality software applications. Computer failure prediction is a key aspect to improve the efficiency of software applications. Computer reliability is a model that can accurately predict software failures. Machine Learning (ML) solves various problems by developing programs that can enhance the output of specific processes based on existing knowledge. These algorithms are very useful for developing various forms of applications. This Machine Learning Algorithm focuses on a variety of well-known problem areas in which humans lack the necessary skills to create successful algorithms. Due to the conversion of certain software tasks into learning problems and learning algorithms, the production of programs becomes more tedious. the Rayleigh Model is combined with The Naive Bayes Algorithm (NBA) to identify software defects, and the proposed Quality Centered Method is used to verify the consistency of the software.

Index terms: Software Engineering (SE), Machine Learning (ML), Quality-Centered Method (QCM).

I. INTRODUCTION

Software quality is one of the important factors in software development. Software reliability is widely used to improve the quality of software applications. Software defect prediction is mainly used to predict defects in software applications. Software defects have a greater impact on software development, and these defects have a greater impact on software cost and maintenance [1]. Developing defect-free software can provide more accurate performance. This will make customers and developers feel more comfortable. Even without a software reliability model, it is difficult to develop defect-free software applications, because most defects are hidden errors obtained after software application development. Defect prediction is the most important activity in software development. Predicting defective modules before software deployment can satisfy customers and improve overall software performance.

In this paper introduces a Quality-Centered Method (QCM) to overcome various problems in existing methods. QCM is composed of Rayleigh model and Naive Bayes algorithm, which can improve software reliability, thereby improving defect prediction in software applications.
II. BACKGROUND WORK

Sharma and Chandra [2] incorporated the latest findings of each machine learning method and the latest advances in equipment error prediction using machine learning into their analysis. This research will be the basis or first step of potential research in software defect prediction. NB is a fast and easy probability classifier based on Bayes’ theorem and the assumption of function independence. The presence or absence of certain types of features has nothing to do with the presence or absence of certain other types of features [3].

Peng He et al. performed an observational analysis using a simplified metric set to forecast software defects [4]. The study compared 34 updates of ten open source projects hosted at the PROMISE database. The finding indicates because when compared to standard predictor variables, the quality of top-k parameters or a minimum metrics subset is acceptable. The reduced or limited set is effective whenever limited resources are necessary.

Grishma BR et al. investigated the root cause of fault detection using clustering techniques and recognized defects to occur and during SDLC’s various phases.

Anuradha Zug and others. Three directional learning calculations and three independent [clustering] learning calculations are used to predict abandonment in programming. The NASA MDP data set is controlled using Weka equipment.

Jaechang Name et al. applied Heterogeneous Defect Prediction [HDP] to predict deserts within and between projects with various data sets.

According to Ebubeogu et al. The absolute number of product defects can be predicted by index factors such as deformation thickness, defect speed and defect presentation time obtained by increasing the deformation speed. The machine learning heuristic [MACLI] method is used to predict deserts. The proposed defect prediction system has two stages. 1) Data pre-processing. 2) Data verification [9].

III. RAYLEIGH MODEL

The uniqueness of the Rayleigh model is that it focuses on the early recognition and defect processing of past object recognition. According to this model, reducing the false injection rate will reduce the area under the Rayleigh bend, resulting in a lower expected field distortion rate. It is ideal to find most errors before an organized exam, because a more appropriate exam will prompt more on-site absconding. The ice inspection is to clarify the synergy between prospecting and on-site disappointment rate, which is just a bigger reminder. In this case, the height of the ice cover is related to the number of incorrect injections. At the beginning of normal operation, the ice shelf may have matured and the size of the ice sheet has been determined. The bigger the tip,
the bigger the ice sheet. For flooded ice, further efforts should be made to open several ice shelves for water supply. The interaction starts from the various stages of investigation, planning, coding, testing, and maintenance. The recognition of each deformity depends on the acuity of the stage. If there is not enough opportunity to assign tasks to representatives, the errand should be completed by other workers in another stage. If the task is completed, this may affect cost and quality. Quality forecasts that rely on early information are in sharp contrast to the last indicator at the end of the advancing cycle. In any case, the desert being evaluated is most likely to prove the quality of the process and can therefore be moved easily.

**Deriving Rayleigh’s Defect Prediction**

Rayleigh expects that two sources of information are needed: the number of scheduled stages and the absolute start-to-end abandonment. Then, the model predicts the number of defects at each stage at that point. Rayleigh work is given by the following formula:

\[
\text{Estimated Defect Density in Phase} = E \times (e^{-B(p-1)^2} - e^{-Bp^2}) \quad \text{(Equation-1)}
\]

Where \( E \) = Total no. of Defects

\( P \) = Phase Number

\( B \) = Defect Detection Efficiency

\( D \) = No. of Defects Found in a Phase

\( T \) = Total Number of Defects in the Phase

Defect Detection Efficiency (DDE) can be calculated by using the following method:

\[
B = \frac{D}{T} \times 100 \quad \text{(Equation - 2)}
\]

The parameters that are used to calculate the performance of the proposed Rayleigh model by using equation (1) and (2) by using following equations.

**Naive Bayes Classifier**

The Naive Bayesian Prediction (NBP) model selects software modules as the target unit for training and prediction. The software module is: 1) a program unit, which is discrete and identifiable in terms of compilation, combination with other units, and loading; 2) a logically separable part of the program. Let \( A = \{a_1, a_2, an\} \) be the set of metric attribute sets, and there is a vector \( M: \{(a_1, w_1), (a_2, w_2), (an, wn)\} \) Represents the software module, where \( ai \) is an indicator and \( wi \) is the weight of \( ai \). We use the module data set with category labels to train the classifier and calculate the defect probability of the new module. When a new module exceeds the threshold, an alarm will be issued. If we define the category concept of the software module as \( c \in \{Cd, Cn\} \), where \( Cd \) is the defect category and \( Cn \) is the non-defect category. Then according to Bayesian theory [13], the probability of software module failure will be calculated by (3):

\[
P[C_d|M] = \frac{P[M|C_d] \times P[C_d]}{\sum_{c \in C} P[c] \times P[M|c]} \quad \text{Equation (3)}
\]

As shown in (4) from the above equation the modules are classified as \( M \) to \( C_d \) when the ratio is greater than \( \lambda \).

\[
\frac{P[C_d|M]}{P[C_n|M]} > \lambda \quad \text{Equation (4)}
\]

\( P[M|c] \) and \( P[c] \) will predict using training data sets. Where \( P[c] \) will be calculated by (5):

\[
P[c] = \frac{N_c}{N} \quad \text{Equation (5)}
\]
Software Cost Estimation (SCE)

It is the most significant factor for every software project. It is very important to know the development cost and time taken to complete the total project. All these estimations are needed before the development is to be started. But the overall software cost is calculated based on the efforts of the software developers. Software cost is totally based on effort of every project member. Software defects show the more impact on software cost. The following equations show the cost calculation. The cost estimation models are such as static and dynamic models are present. The static model is based on key factor that calculates the cost and time. For the dynamic model, there is no default variable, all the variables are interdependent.

\[ C = xS^y \]  

Where C=Cost, S-Size of Code, x and y are variables.

Quality Centered Method (QCM)

QCM is a hybrid algorithm consisting of Rayleigh defect prediction and Naive Bayes classifier algorithm. Naive Bayes (NB) is a powerful artificial intelligence strategy. The NB model treats incomplete expectations as a pair of orders, prepares and constructs indicators by checking the verifiable information of the product module, and judges whether there is desert in the new module based on the indicators. The test results show that the performance of NB is inherently better, which proves the feasibility and significance of NB in software defect prediction.

Dataset Description

The data set used is, the synthetic software defects data with unlimited random generation of data. Two projects are selected with time bound and named as project-1, Project-2 and Project-3. This data can be used in any of the software organization to improve the quality of the software projects or products. The software size i,e the dataset size is according to the time bounding limits. The parameters such as Analysis for Defect, Quality & Cost Predictions, and total time for processing are shown in figure 2, 3, 4.

![Software Phase Data](image)

Figure: 1 Analysis of the software defect prediction for project-1 based on phases

![Software Phase Data](image)

Figure: 2 phase wise defect prediction analysis for project-1 using Rayleigh model and Naive Bayes Algorithm

Start: 2020-12-01T00:00:00.000+05:30
The analysis of project-1 with proposed methodology shows the phase wise analysis of data generated by the software company. Analysis done based on generated data.

**Software Phase Data**

*Analyzer Initiated @ 2021/02/14 17:24:32*

<table>
<thead>
<tr>
<th>WorkLoad Scheduling Across SDLC Teams</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyzer's Size</td>
<td>18</td>
</tr>
<tr>
<td>Designer's Size</td>
<td>12</td>
</tr>
<tr>
<td>Developers's Size</td>
<td>147</td>
</tr>
<tr>
<td>Tester's Size</td>
<td>61</td>
</tr>
<tr>
<td>Deployer's Size</td>
<td>7</td>
</tr>
<tr>
<td>Total's Size</td>
<td>245</td>
</tr>
<tr>
<td>Work Days Allocated</td>
<td>123</td>
</tr>
<tr>
<td>Work Days Scheduled</td>
<td>105</td>
</tr>
</tbody>
</table>

Figure: 4 Analysis of the software defect prediction for project-2 based on phases
Figure: 5 phase wise defect prediction analysis for project-2 using Rayleigh model and Naive Bayes Algorithm

<table>
<thead>
<tr>
<th>SDLC Stage</th>
<th>Previous Version Defects</th>
<th>Predicting Probable Current Version Defects</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>159</td>
<td>246</td>
<td>0.0</td>
</tr>
<tr>
<td>Design</td>
<td>45</td>
<td>270</td>
<td>0.0</td>
</tr>
<tr>
<td>Coding</td>
<td>449</td>
<td>762</td>
<td>0.0</td>
</tr>
<tr>
<td>Testing</td>
<td>830</td>
<td>606</td>
<td>0.0</td>
</tr>
<tr>
<td>Maintenance</td>
<td>29</td>
<td>36</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Defect Conclusion : 0.0 Defects Prediction (Quality Centered Method(QCM))

Cost Conclusion : 0.0% Budget To Build Current Software

Start: 2020-10-01T00:00:00.000+05:30
End: 2021-01-31T00:00:00.000+05:30
Processing Time: 14.3352678 seconds
IV. CONCLUSION

In this paper, the Quality-Centered Method (QCM) is developed to improve the quality of the software by detecting the defects in the specific time domain. Rayleigh is the software reliability model which is used to predict the defects in the phase wise. Naive Bayes’ algorithm is also integrated to improve the performance of the Rayleigh model to overcome the issues in defect prediction. This will improve the quality of the software project.

In future, a hybrid model is required to improve the software defect prediction with the integration of machine learning algorithm. Effective defects prediction can show impact on software development and cost reduction.

REFERENCES