WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

EFFICIENT SOFTWARE DEFECTS PREDICATION USING SVM

Prof. Aparna Patil, Vijay Kasar, Abhishek Gaikwad, Ashish Mahajan, Abhishek Khodke.

Computer Department, Genba Sopanrao Moze Collage Of Engineering, Balewadi, Pune.

ABSTRACT

Many Organization wants to detect the number of faults in software system before they are placed, to scale the likely delivered quality and maintenance efforts. This helps many software and qualitative models have been introduced with a correspondingly great literature. We give a serious review of this literature. The bulk of broad range of forecast models use size and complexity metrics to detect defect. Other software's are depend on testing data, the "grade" of the development process, or take a multiple viewpoint(approach). Software defect prediction gives development groups with noticeable results, while giving to industrial outcomes and development defects detecting faulty code section can also help developers to recognize bugs and maintain their test activities different methods are used to identify or detect the classification part, the most known Machine Learning Algorithms like Support Vector Machine(SVM), Decision Tree(DT), Logistic Regression(LR). A organized research analysis is managed in framework of confusion, precision, recall, reorganization, accuracy, etc. The systematic analysis shows that the purpose outcome will give more useful solution for software defect prediction.

***Keywords**: Defect Prediction Software; Metric Software's; Machine Learning; Complexity Metrics; SVM; DT; LR*

1. INTRODUCTION

In the Past few years, people have constantly focus on software quality software base system in which software quality is consider as a most critical element in user capability. The main goal of software defect prediction is to detect the fault in software hence to minimize the efforts and time taken as well as cost and grade of software the machine learning algorithm is used for the training and testing this purpose Support Vector Machine (SVM) algorithm. The characteristic of classification known as the relationship between attributes and training dataset is generated on the classifier method and inspect through the classification of the targets defining software faults, finding the faults and recognizing it is a constant work for researchers because of huge distribution of software.

The main goal of classifying the software data set as a model for error prediction into a defective and non-defective data set. The input software data set is given to categorizer in consonance with this method where the user knows the real class values. Requirement base and design based metric methods illustrate significant results before this scheme but the structure of algorithms and correctness of predictions remain a difficult task.

2. RELATED WORK

Machine learning may be a powerful methodology for prediction, software system defect prediction model projected by Wang et al. for increasing the amount of application software systems. Databases of defective software system comprise of unbalanced information that produces random patterns. This problem encourages the creation of a good and reliable classifier of things for educational and industrial applications. Xu et al. researched "software defect prediction ways and hypothesized that ancient techniques use vectorization and have selection" framework to minimize trivial options, however still exclude alternative essential features leading to degraded performance of defect prediction strategy. "utilized a well known methodology in machine learning, i.e. SVM (support vector machine). Besides, predictability in attributes is mentioned through the diligence of a domestically linear embedding strategy with a support vector classifier. SVM constraints square measure so designed with a tenfold cross-validation method and grid search theme according to this approach".

"Support Vector Machine" (SVM) may be a supervised ML Algorithm that are often used for each classification or regression challenges. However, it's principally utilized in classification issues. within the SVM rule, we tend to plot every information item as some extent in n-dimensional house (where n may be a variety of options you have) with the worth of every feature being the worth of a selected coordinate.

3. LITERATURE SURVEY

Previous research approaches on software defect prediction has taken the researchers to one of these two directions: one is creating new features or using combinations of existing features for better accuracy. Scientists and researchers have designed hand-crafted metrices to detect defect characteristic for example, Halstead features based on operator and operand counts, McCabe features based on dependencies, and other comprehensive feature sets including metrics for object-oriented design features, and code change features. As for the second direction, machine learning techniques, as many machine learning models have been designed to classify data for example, decision tree, Support vector machine, Logistic regression, and dictionary learning models. Those approaches however do not sufficiently capture the syntax and different levels of semantics of source code as software metric features work well in some projects and may not perform well in other projects.

4. PROBLEM STATEMENT

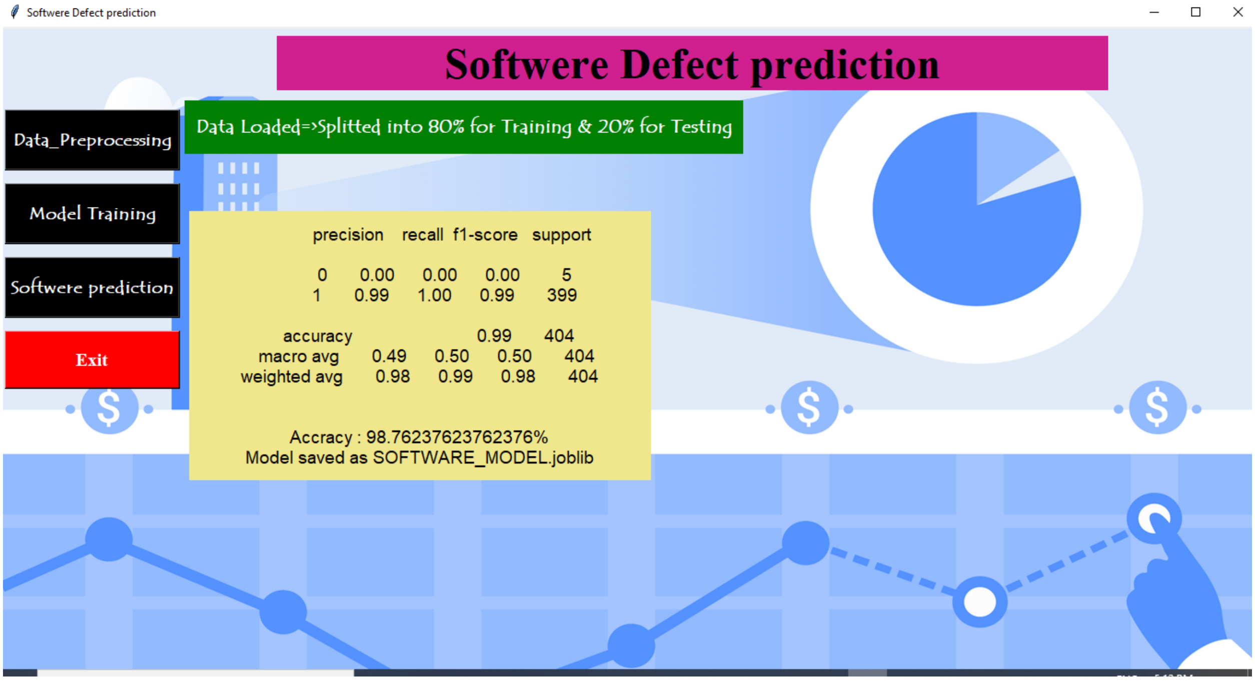
It is also necessary to test thoroughly in order to fulfil and meet customer demands, even if this is costly and time-consuming. Some modules may be defective, while others may be defect-free. If the minority modules contain defects, it is not necessary to test all of them.

GOALS AND OBJECTIVES:

- The main objective of this project is Increasing confidence in the level of quality and providing information about it. ensure that the final product fits the needs of both the business and the users.
- Without breaching the restrictions, the testing resources can be employed effectively.
- Identifying faults that may have been introduced by the programmer during the development of the software.
- Developing and disseminating information regarding the level of quality. To eliminate flaws. To ensure that the final product fits the needs of both the business and the users.
- Software reliability is frequently underestimated until after development has been completed. The true reliability of the system will not be known until it is delivered to clients. Later in the process, corrective action is more costly.
- Gaining confidence in and providing information about the level of quality.
- To make sure that the end result meets the business and user requirements.
- the testing resources can be used efficiently without violating the constraints.

STATEMENT OF SCOPE:

Software defect prediction has been an inspiration to software engineers as scale and complexity of projects are increasing with modern technology, defects will increase. Defect prediction has been implemented using different well-known machine learning algorithms that has performed well in software defect predictions, however it was not sufficient to capture the syntax and different level of semantics. In this Project we use Support Vector Machine(SVM), Decision Tree(DT), Logistic Regression(LR) Algorithms. When we use SVM we got have observe that SVM showing less accuracy, so we filter prediction with decision tree algorithm and Logistic Regression algorithm for predicting software defect with more accuracy.



Data Loaded=>Splitted into 80% for Training & 20% for Testing

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         5
           1       0.99      1.00      0.99       399

    accuracy                           0.99       404
   macro avg       0.49      0.50      0.50       404
weighted avg       0.98      0.99      0.98       404


Accracy : 98.76237623762376%
Model saved as SOFTWARE_MODEL.joblib
```

METHODS:

A) SVM (SUPPORT VECTOR MACHINE) a] Support vector machines are a set of supervised learning methods used for classification, regression, and outliers detection. b] Import the dataset. c] Explore the data to figure out what they look like. d] Pre-process the data. e] Split the data into attributes and labels. f] Divide the data into training and testing sets.

B) DECISION TREE: - a] Decision tree are set of supervised learning method. b] It is used for both Classification and Regression problems, but mostly it is preferred for solving Classification problems. c] There are two nodes, the Decision Node and Leaf Node. d] Decision node is used to make decision and Leaf node are the output of those decisions. e] It is a graphical representation for getting all the possible solutions to a problem based on given conditions.

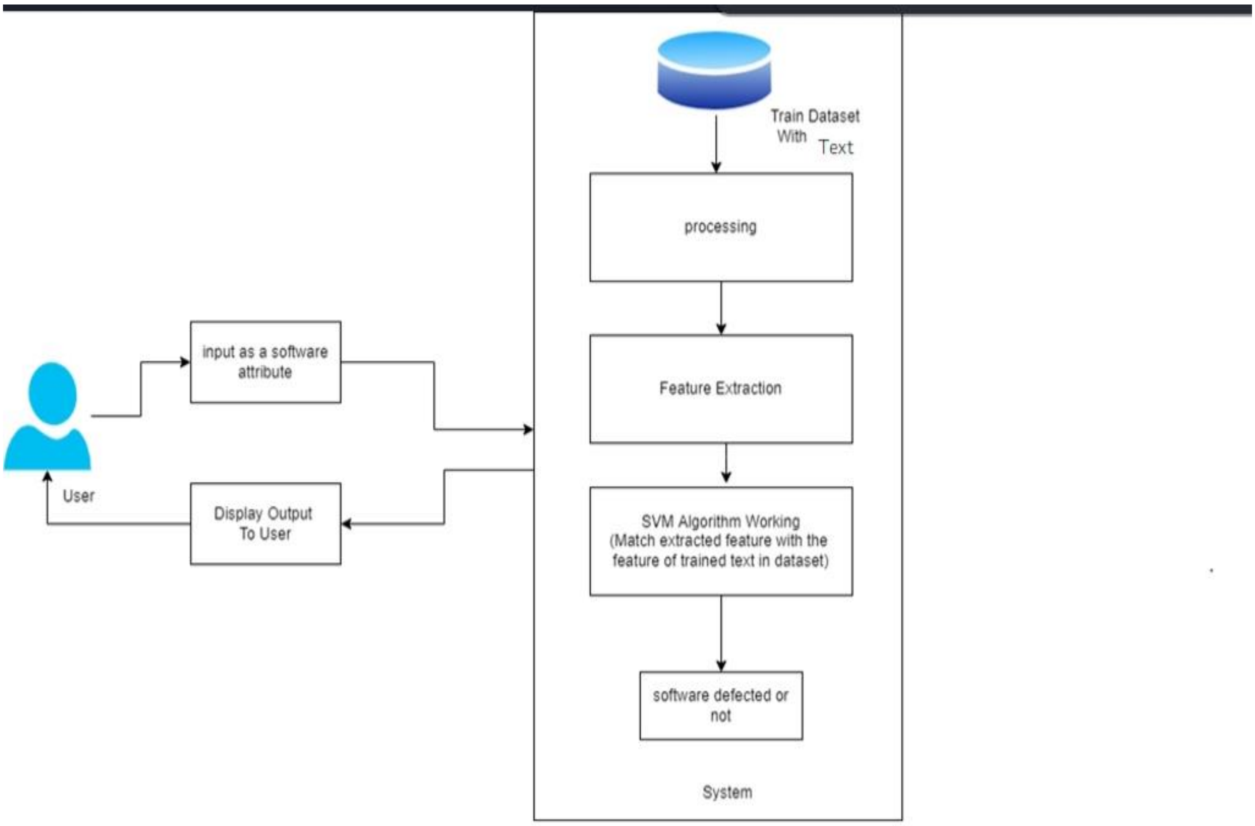
C) LR (Logistic Regression): a] Logical Reasoning are set of supervised learning method. b] It is used for solving the classification problems. c] It is used to predict the probability of the targeted value. d] The output must be categorical or discrete value that it either can be 1 or 0, YES or NO.

SYSTEM WORK AS FOLLOWS:

Admin will import the dataset, extract the features and train the module.

- The model is train on based on different parameters like cyclomatic complexity, essential complexity, program length, etc.
- The system will provide the accuracy metrics of SVM.
- It will load the GUI, user will login and enter the parameters.
- Then based on input parameter the system will predicts that there is defect or not. If there will be defect on based on input parameters then is display software defect detected and if no defect is present then it will display no software defect detected.

ARCHITECTURE DIAGRAM:-



FEATURES:

- The GUI page will load.
- User registration: - User has to register himself and create a unique user name and password.

- User login: - User login using his user name and password.
- Check software defect: - User will enter the given parameters of the software in respective tabs.

5. CONCLUSIONS

This paper centered on scrutiny the foremost well-known machine learning algorithms that are wide accustomed predict computer code defects. For the coaching And Testing we have a tendency to Use SVM algorithmic rule. Through the applying we have a tendency to simply notice Defect computer code. Much of the revealed empirical add the defect prediction area is well beforehand of the unsupported rhetoric sadly typical of abundant of what passes for computer code engineering research. but each discipline should learn the maximum amount, if not more, from its failures as its successes. during this spirit we tend to have reviewed the literature critically with a read to higher understand past failures and description attainable avenues for future success.

6. FUTURE WORK

In future we will apply some other machine learning algorithms for Improving result accuracy, And will try to update the features and range of our software in order a organization can use it.

REFERENCES

[1] CB-Path2Vec:ACrossBlockPathBasedRepresentationforSoftwareDefect Prediction .XiyuZhang1*,YangLu1,2,KeShi

[2] Study on Software Defect Prediction Model Based on Improved BP Algorithm .Cundong Tang1,2,*, Li Chen1, Zhiping Wang2, Yuzhou Sima2

[3] Connecting Software Reliability Growth Models to Software Defect Trackin .Maskura Nafreen, Melanie Luperon, and Lance Fiondella

[4] BRACE: Cloud-based Software Reliability Assurance Kazuhira Okumoto Software and Systems Reliability Engineering Bell Labs CTO, Nokia Dublin, Ireland Kazu.okumoto@nokia-bell-labs.com Abhaya Asthana Software and Systems Reliability Engineering Bell Labs CTO, Nokia Westford, USA abhaya.asthana@nokia-bell-labs.com Rashid Mijumbi Software and Systems Reliability Engineering Bell Labs CTO, Nokia Dublin, Ireland

[5] CB-Path2Vec:ACrossBlockPathBasedRepresentationforSoftwareDefect Prediction .XiyuZhang1*,YangLu1,2,KeShi

[6] Study on Software Defect Prediction Model Based on Improved BP Algorithm .Cundong Tang1,2,*, Li Chen1, Zhiping Wang2, Yuzhou Sima2

[7] Connecting Software Reliability Growth Models to Software Defect Trackin .Maskura Nafreen, Melanie Luperon, and Lance Fiondella

[8] BRACE: Cloud-based Software Reliability Assurance Kazuhira Okumoto Software and Systems Reliability Engineering Bell Labs CTO, Nokia Dublin, Ireland Kazu.okumoto@nokia-bell-labs.com Abhaya Asthana Software and Systems Reliability Engineering Bell Labs CTO, Nokia Westford, USA abhaya.asthana@nokia-bell-labs.com Rashid Mijumbi Software and Systems Reliability Engineering Bell Labs CTO, Nokia Dublin, Ireland