Software Defect Prediction Using Machine Learning

P.L.S.Tejaswini, K.Sree Varsha, P.Yasaswini, Sangeetha Yalamanchili

Abstract—Software defect prediction analysis is an important problem in the software engineering community. Software defect prediction can directly affect the quality and has achieved significant popularity in the last few years. This software prediction analysis helps in delivering the best development and makes the maintenance of software more reliable. This is because predicting the software faults in the earlier phase improves the software quality, efficiency, reliability and the overall cost in SDLC. Developing and improving the software defect prediction model is a challenging task and many techniques are introducing for better performance. Supervised ML algorithms have been used to predict future software faults based on historical data[1]. These classifiers are Naïve Bayes(NB), Support Vector Machine(SVM) and Artificial neural network(ANN). The evaluation process showed that ML algorithms can be used effectively with a high accuracy rate. The comparison is made with other machine learning algorithms to finds the algorithms which give more accuracy. And the results show that machine learning algorithms gives the best performance. The existence of software defects affects dramatically on software reliability, quality, and maintenance cost. Achieving reliable software also is hard work, even the software applied carefully because most time there is hidden errors. In addition, developing a software defect prediction model which can predict the faulty modules in the early phase is a real challenge in software engineering. Software defect prediction analysis is an essential activity in software development. This is because predicting the bugs prior to software deployment achieves user satisfaction, and helps in increasing the overall performance of the software. Moreover, predicting software defects early improves software adaptation to different environments and increases resource utilization.

Keywords: Naïve Bayes, Support Vector Machine(SVM), Artificial Network Network(ANN), Software Defect Analysis, Reliability, Anaconda Navigator.

I. INTRODUCTION

Software Defect Prediction is an important issue in software development and maintenance processes, which concerns with the overall of software success. Predicting and finding the bugs in the earlier phase in SDLC makes the software more reliable, efficient and better quality when compared with finding bugs in the later stages. However, developing a software defect prediction model is not an easy task and many new tools and methods are introducing in the machine learning for better performance. These classifiers are Naïve Bayes(NB) and Support Vector Machine(SVM) and Artificial Neural Networks(ANN). The development procedure demonstrated that ML calculations can be utilized adequately with a high precision rate. A programming deformity is a blunder, bug, imperfection, issue, breakdown or errors in programming that makes it make a mistaken or unpredicted result. Issues are basic properties of a framework. They show up from structure or assembling, or outside condition. Programming blemishes are customizing mistakes which cause distinctive execution contrasted and expectation. The dominant parts of the flaws are from source code or plan, some of them are from the mistaken code producing from compilers. For programming designers and customers, programming deficiencies are a perilous issue. Programming abandons not only diminish programming quality, increment cost yet in addition postpone the advancement plan. Programming deficiency anticipating is proposed to settle this kind of trouble[4].

II. APPLICATIONS

This Software defect prediction helps in deploying the products which are defect free and satisfy the customers. It helps in reducing the cost of testing the software. It helps to identify the errors in the early stages. In less time we can find the number of bugs. Helps to deploy the products earlier than expected time.

III. METHODOLOGY

Pre Processing:

We need to do the preprocessing of the data so that the features of the data can be extracted. After preprocessing load the data into the next step

Machine Learning:

Machine learning is one of the applications in Artificial Intelligence. It helps the system to learn from experience and through programming, and it also helps in improving their ability to learn. The main aim of machine learning is to develop programs for the systems so that they learn by accessing the given data.
**Supervised Machine Learning:**

Managed learning, with regards to computerized reasoning (AI) and AI, is a kind of framework where both input and wanted yield information are provided[3]. Information and yield information are named for arrangement.

**IV. CLASSIFICATION**

Classification is nothing but identifying the categories to which the given data set belongs to by training the given data with instances having different categories.

**Naïve bayes:**

Naïve Bayes is one of the order methods which depends on Bayes Theorem. Gullible Bayes classifier accept that the nearness of any component in a class isn't identified with the nearness of some other element.

space and mapping them to find the class and they are separated by a line having the gap.

**Gaussian Naïve Bayes :**

A Gaussian Naïve Bayes algorithm is a special type of Naive Bayes algorithm. It is used when the features have continuous values. It's also assumed that all the features are following a Gaussian distribution i.e, normal distribution.

**Support Vector Machine:**

support vector machines (SVMs) is a directed learning strategy. SVM gains from preparing the dataset and it is utilized for grouping. On the off chance that we think about a lot of preparing models, each case having a place with one of two classes, an SVM calculation assembles a model that aids in anticipating whether the model falls into one class or the other one. Basically, we can think about an SVM model by speaking to the model in

**Artificial Neural Network:**

Above,

Artificial Neural Networks (ANN) helps in designing the system to simulate the way, how the human brain analyzes and processes the given information. This technology helps in solving the problems that are impossible to solve by human brains and that beyond the imagination.

\[ P(h|d) = \frac{P(d|h)P(h)}{P(d)} \]

Above,

Artificial Neural Networks (ANN) helps in designing the system to simulate the way, how the human brain analyzes and processes the given information. This technology helps in solving the problems that are impossible to solve by human brains and that beyond the imagination.

\[ P(h|d) \text{ is the posterior probability of class (c, target) given predictor (x, attributes).} \]

\[ P(h) \text{ is the prior probability of class.} \]

\[ P(d|h) \text{ is the likelihood which is the probability of predictor given class. P(d) is the prior probability of predictor.} \]

**Datasets:**

The dataset contains the historic data of the software application contains defects. It has attributes that have an effect on the application. The features in the datasets were defined in the 70s in an attempt to objectively characterize code features that are associated with software quality.
V. DESIGN METHODOLOGY

Here in this design methodology, we represent the basic design methodology for the steps involved in the process of software defect prediction analysis. These are the main important steps involved and are explained in the system architecture.

The first step we need to do is to select the datasets with software defects. The Datasets we collected are the benchmark datasets. And select the algorithms that we need to implement. Here we choose the best machine learning algorithms Naïve Bayes, SVM, and ANN. Apply to preprocess the data so that any missing values can be replaced with other values or it can be used for the feature extraction. New apply the classification so that the data is converted into 0’s and 1’s by applying binary classification. And at last, find the accuracies by using Naïve Bayes and SVM and ANN algorithm to the compute the accuracies. And we predict the algorithm that gives the most accurate results.
VI. EXPERIMENTAL RESULTS: NAIVE BAYES
Calculate the accuracy of the datasets using Naive Bayes

SVM:
Calculating the accuracy using SVM

NAIVE BAYES AND SVM:
Comparison of accuracies between Naive Bayes and SVM

Accuracies of all the datasets of Naive Bayes and SVM

0.8211570247033985

ANN:
Two-layer perceptron based neural network for all the data
Training_Set...Overall: 12098
Testing_Set...Overall: 3025
Training Done!
The accuracy for Overall test batch is 0.817851
Plotting the training loss for overall project using two-layer Perceptron

Two-layer perceptron based neural network for CM1
Training_Set: 398
Testing_Set: 100
Training Done!
The accuracy for CM1 test batch is 0.85
Plotting the training loss for CM1 using two-layer perceptron
Changing Hidden layer size and checking Accuracy for CM1
Training Done!
The accuracy for CM1 test batch is: 0.85 Current Hidden Layer Size: 19
Training Done!
The accuracy for CM1 test batch is: 0.85 Current Hidden Layer Size: 20
Training Done!
The accuracy for CM1 test batch is: 0.85

Plot Accuracy-vs-Hidden Layer Increments for CM1

Changing Hidden layer size and checking Accuracy for Overall data
Current Hidden Layer Size: 3 – Training Done!
The accuracy for our test batch is: 0.816859 Current Hidden Layer Size: 4
Training Done!
The accuracy for our test batch is: 0.815868 Current Hidden Layer Size: 5
Training Done!
The accuracy for our test batch is: 0.817521 Current Hidden Layer Size: 19
Training Done!
The accuracy for our test batch is: 0.81719 Current Hidden Layer Size: 20
Training Done!
The accuracy for our test batch is: 0.818843

Plot Accuracy-vs-Hidden Layer Increments for Overall Data

VII. CONCLUSION
Software Defect Prediction can directly affect the quality and has achieved significant popularity in the last few years. This software prediction analysis helps in delivering the best quality product without any defects. Therefore this helps in deploying the products that are error free. Here we performed this using machine learning algorithms Naive Bayes, Support vector machine and Artificial Neural Networks. When we observed the accuracies obtained between these algorithms SVM is more accurate than Naive Bayes and Artificial neural network (ANN) is more accurate than both of NB and SVM.

REFERENCES
1. http://www.ijceccom/comparative-study-analysis-classification-algorithms-machine-learning/on"COMPARATIVE STUDY AND ANALYSIS OF CLASSIFICATION ALGORITHMS THROUGH MACHINE LEARNING" by Dr. V. T. Meenatchi, Dr. M. Thangaraj, Dr. V. Gayathri, Dr. S. Gnanamal