International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-10 Issue-1, May 2021

Software Defect Estimation using Machine Learning Algorithms

Revoori Veeharika Reddy, Nagella Kedharnath, Mandi Akif Hussain, S. Vidya

Abstract: ***Software Engineering is a branch of computer science that enables tight communication between system software and training it as per the requirement of the user. We have selected seven distinct algorithms from machine learning techniques and are going to test them using the data sets acquired for NASA public promise repositories. The results of our project enable the users of this software to bag up the defects are selecting the most efficient of given algorithms in doing their further respective tasks, resulting in effective results.***

Keywords: Software Quality Metrics, Software Defect Prediction, Software Fault Prediction, Machine Learning Algorithms

I. INTRODUCTION

A) Problem Statement:

Now-a-days developing software system is a difficult process which involves planning, analysing, designing, implementing, test, integrate and maintenance. A software engineer work is developing a system in time with limited budget which is done in planning phase. While doing the development process we can have few defects like not proper design, where the logic is poor, data handling is improper, etc. and these defects cause errors which lead to re-do the work, increasing in development and cost of maintenance. This all are responsible for the decrease in customer satisfaction. In this point of view, faults are grouped on the basis of sternness, corrective and advance actions are taken as per the sternness defined. The selected machine learning algorithms for comparison are

(i) Bagging
(ii) Random Forests (RF)
(iii) Multilayer Perceptron (MLP)
(iv) Radial Basis Function (RBF)
(v) Naive Bayes
(vi) Multinomial Naive Bayes
(vii) Support Vector Machine (SVM).

B) Objective:

The Objective of this project is to estimate the defect of software using machine learning algorithms.

Manuscript received on May 17, 2021.
Revised Manuscript received on May 20, 2021.
Manuscript published on May 30, 2021.
* Correspondence Author
Mandi Akif Hussain*, Pursuing, Graduation in Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnankoil (Tamil Nadu), India.
Revoori Veeharika Reddy, Pursuing, Graduation in Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnankoil (Tamil Nadu), India.
Nagella Kedharnath, Pursuing, Graduation in Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnankoil (Tamil Nadu), India.
S. Vidya, Associate Professor, Department Computer Science, Kalasalingam Academy of Research and Education, Krishnankoil (Tamil Nadu), India.

On training the various ML Algorithms we need to get good accuracy percentage so that the particular algorithm fits the best in order to estimate the defects Support Vector Machine (SVM) supports both classification as well as regression. It is productive and straight-lined method which is used in classification. For classification it divides two groups by making boundaries between the group of data.

C) Proposed System:

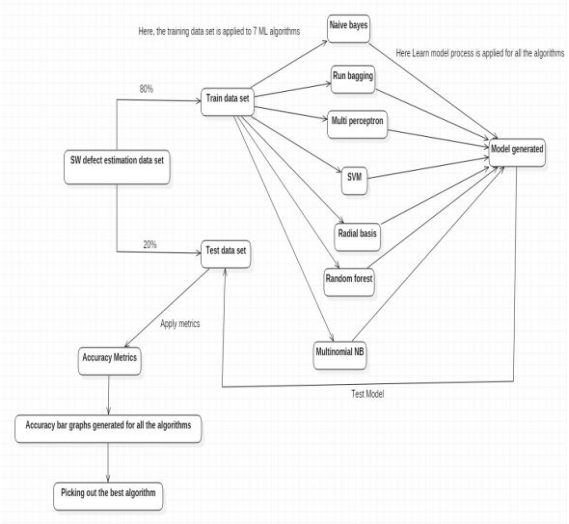
The proposed system includes SVM, Multilayer Perceptron, Run Bagging algorithm, Naive Bayes algorithm, Random Forest algorithm, Multinomial NB and Radial Basis Functions to solve the class misbalancing problem which causes in the decreasing performance of defect prediction. The dataset has been trained and spitted according to the constraints and using the accuracies has been defined in order to measure the defect estimation capability of various algorithms proposed.

D) Advantages of proposed system:

1. Predicted model is used for evaluating the performance measures.
2. We can apply various datasets in this project. But we are using NASA datasets in our project.
3. Software defects are classified to the extent.
4. Advance measures can be taken on selection of algorithm
5. Provides Better results.
6. Identify defects in the early stage of the project which in turn results in Customer loyalty.

II. SYSTEM DESIGN

A) System Architecture:



Retrieval Number: 100.1/ijrte.A58980510121
DOI: 10.35940/ijrte.A5898.0510121
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

B) Module Description:

Interface Designing Module:

The main page of the project is developed using Tkinter GUI.

Tkinter is the standard GUI library for Python. Python which is when interacted with Tkinter that provides a speed and simple way to create GUI applications. Tkinter which provides a object-oriented interface to the toolkit Tk GUI.

Tkinter has some standards in it ,they are:

- Dimensions
- Colors
- Fonts
- Anchors
- Relief styles
- Bit maps
- Cursors

C) Use case Diagram:

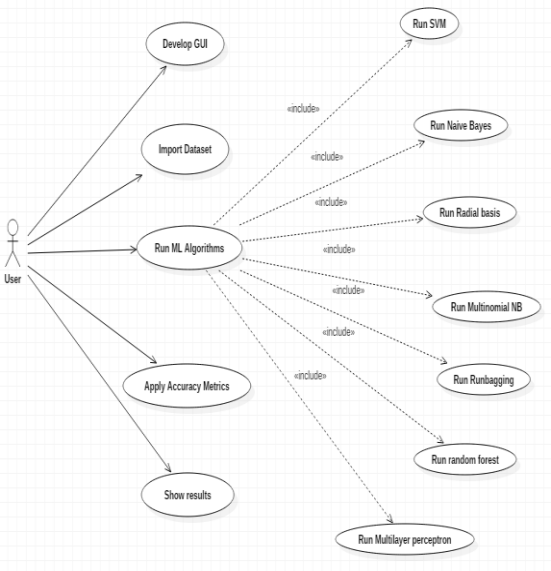
This diagram is a behavioural defined and created form analysis of use case in the language of Unified Modelling Language (UML)

The need of use case is to visualize graphical overview of the capability which is given by a system in the terms of actors, goals and any dependencies among those use cases. The main motive of the figure is to show which system functions are performed for which actor. In the system the roles of the actors can be described.

Use case diagram involves:

- Use cases
- Actors
- Dependencies

By use case diagram one can know the working of the system effectively and understand the dependencies between actor and use case.



D) Sequence Diagram:

It shows how the processes operate with each another and in what order. It is the construction of a Message Sequence Chart. This is explained in Unified Modelling Language (UML).

Notations:

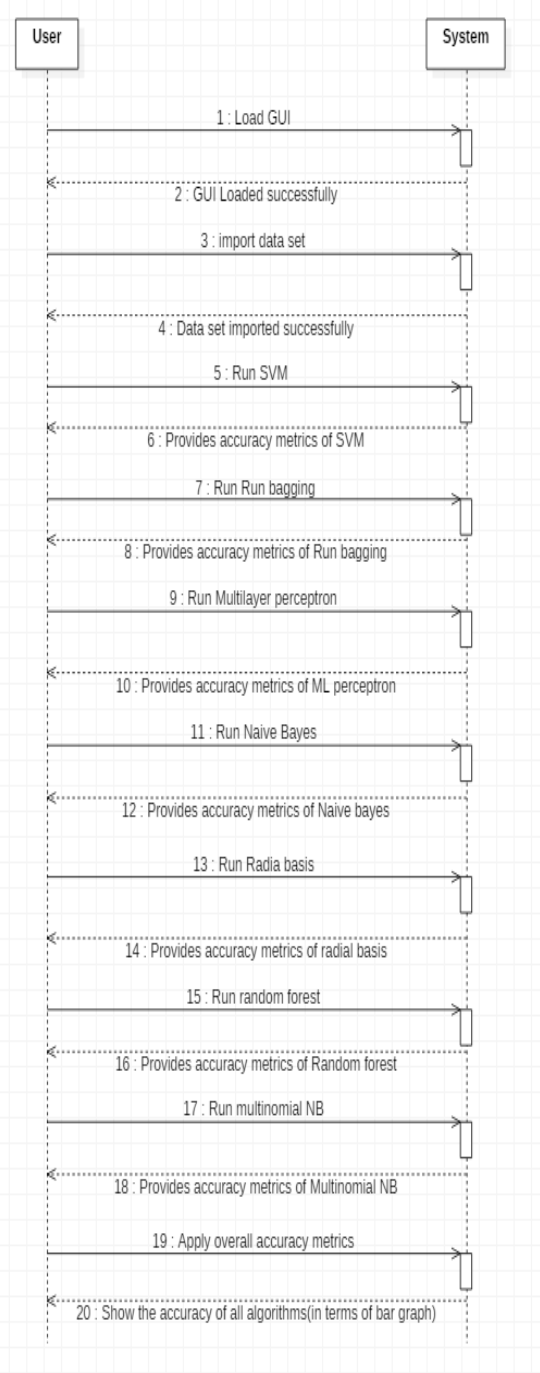
Actors – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.

Lifelines – It is named element which tells about an individual. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

Messages – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

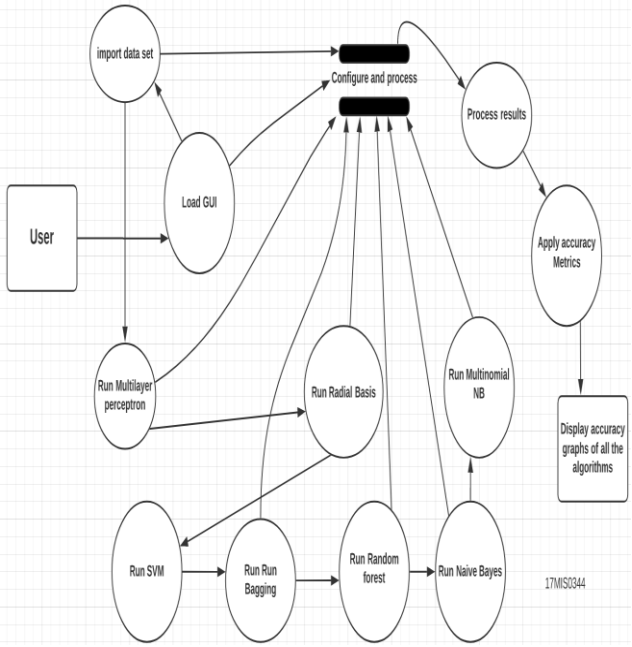
In my project there are two life lines and the message is shared between user and system

Sequence diagram goes as follows:



E) Data Flow Diagram:

- Data flow diagrams are used to represent the graph of flow of data in a business information system. DFD depicted the processes that are involved in a system to transfer data from the input to the file storage and reports generation. These diagrams can be divided into logical and physical.
- The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical DFD describes the implementation of the logical data flow.
- DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system.
- The visual representation makes it a good communication tool between User and System designer. Structure of DFD tells us starting from the overview and expands it to a hierarchy of the detailed diagrams.

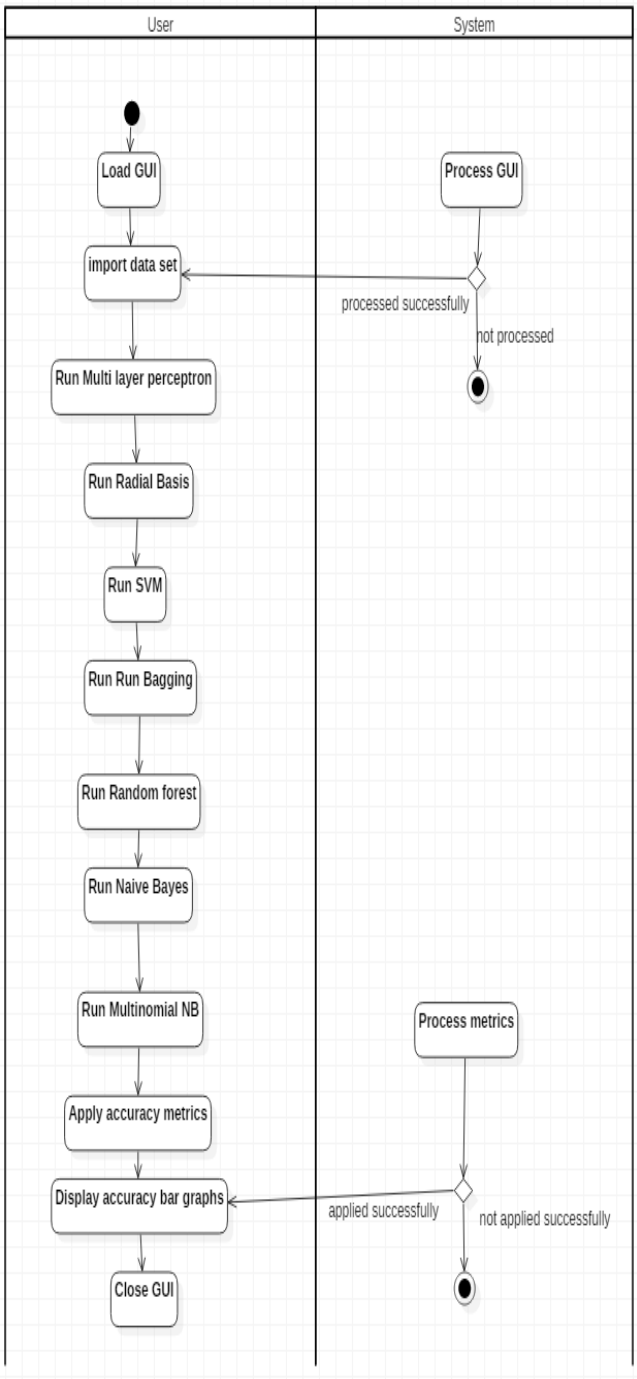


F) Activity Diagram:

It shows the flow of work like graphs in step wise with support for choice, loop and consistency. In the language of unified modelling, activity diagrams are used to describe the business and operational step-by-step workflow of the components in a system. The flow of control is showed here.

- Benefits of activity diagrams
- Activity diagrams present a number of benefits to users. Consider creating an activity diagram to:
- Demonstrate the logic of an algorithm.
- Describe the steps performed in a UML use case.
- Illustrate a business process or workflow between users and the system.
- Simplify and improve any process by clarifying complicated use cases.
- Model software architecture elements, such as method, function, and operation.

- Activity Diagram for my project is as follows: (Used Swim lane) *Swimlanes* are used to show which *activities* are performed by which organisation in the *activity diagram*



III. TEST RESULTS

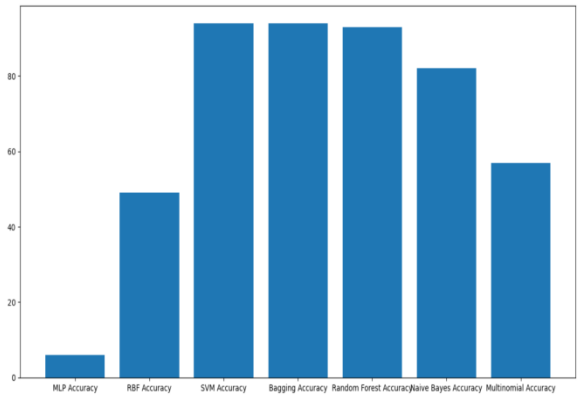
Test case ID	Input	Expected o/p	Actual o/p	Status
1	Loading GUI without clicking on the windows batch file	Display GUI	GUI cannot be loaded	Fail
2	Loading GUI on clicking the windows batch file	Display GUI	GUI displayed	Pass
3	Processing before importing the Data set	Data set imported successfully	Error	Fail
4	Processing after importing the data set	Data set imported successfully	Data set imported successfully and provides the path	Pass
5	Clicking on the "All algorithms accuracy graph" button before executing ML algorithms	Accuracy graph of all the Algorithms	Error	Fail
6	Clicking on the "All algorithms accuracy graph" after executing all ML algorithms	Accuracy graph of all the ML algorithms	Accuracy graph of all the algorithms is displayed	Pass

IV. RESULTS AND DISCUSSIONS

Step-1:
Double click on windows batch file to load the GUI:

GUI:

Step-2:
Click on upload NASA Data Set:
On uploading CM1 Data set we can see total dataset size and training size records and testing size records application obtained from dataset to build train model.
Step-3:
Now click on 'Run Multilayer Perceptron Algorithm' button to generate model and to get its accuracy
Step-4:
Now click on "Run Radial Basis Function Algorithm" button to generate model and to get its accuracy
Step-5:
Click on Support vector Machine Algorithm to generate its accuracy.
Step-6:
Click on Run Bagging Algorithm for generating accuracy.
Step-7:
Click on Run Random Forest algorithm for generating accuracy.
Step-8:
Click on Naive Bayes algorithm for generating accuracy.
Step-9:
Click on Multinomial NB algorithm for generating accuracy.
Step-10:
Click on "All Algorithms Accuracy Graph" Button:



In above graph x-axis represents algorithm name and y-axis represents accuracy of those algorithms.

Retrieval Number: 100.1/ijrte.A58980510121
DOI: 10.35940/ijrte.A5898.0510121
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

B) Discussion:

On seeing the above graphs, we can observe that SVM and Run bagging Algorithms produced better accuracy. So we can hereby proceed on using these two algorithms to overcome the defects of the Software and deploy a quality software to the user. In this project different evaluation metrics are used to evaluate model performance. The graph shows that the ensemble learners are better at software defect estimation and it is also a powerful way to improve the performance of the model. For each ML algorithm and corresponding data set the classification performance result is showed in the text area.

V. CONCLUSION

We are using seven algorithms to predict which is better algorithm in all the seven. The main algorithm which is better is used to detect the defect of the software. So, before releasing it to real environment the defects of the software are predicted by the best algorithm. The best algorithm is chosen while comparing algorithms based on the metrics of software quality which are accuracy, precision etc., the datasets I had taken for this project is CM1 and KC1 which are NASA datasets. They are taken from public repository. By using this datasets and algorithms we choose the best algorithm which saves time while software testing in early phase of cycle. So, we can take actions before they are failures. The consequences of this venture show that tree-organized classifiers as such gathering students which are SVM and Run Bagging have better imperfection expectation execution contrasted with its partners. Particularly, the capacity of Bagging in anticipating programming fault is better. When applied to all datasets, the general exactness, accuracy, review and F-Measure of Bagging is 0.94,1.00,0.97,93. Using these strategies empowers them to save programming testing and support costs by recognizing abandons in the beginning stage of task life cycle and making restorative and preventive moves before they become fall flat.

REFERENCES

1. Felix, Ebubeogu Amarachukwu, and Sai Peck Lee. "Integrated Approach to Software Defect Prediction." IEEE Access 5 (2017): 21524-21547.
2. Chug, Anuradha, and Shafali Dhall. "Software defect prediction using supervised learning algorithm and unsupervised learning algorithm." (2013): 5-01.
3. Perreault, Logan, et al. "Using Classifiers for Software Defect Detection." 26th International Conference on Software Engineering and Data Engineering, SEDE. 2017.
4. Huda, Shamsul, et al. "A Framework for Software Defect Prediction and Metric Selection." IEEE Access (2017).
5. Rajbahadur, Gopi Krishnan, et al. "The impact of using regression models to build defect classifiers." Proceedings of the 14th International Conference on Mining Software Repositories. IEEE Press, 2017.

AUTHORS PROFILE

Mandi Akif Hussain, is currently pursuing graduation in Computer Science and Engineering from Kalasalingam Academy of Research and Education of batch 2017-2021. He has done all of his schooling in his hometown Kurnool, Andhra Pradesh. He was also the Secretary of the Association for Computing Machinery (ACM) for the student chapter of his institution. He was Junior Under Officer (JUO) for the NCC troop of his college

Revoori Veeharika Reddy, is currently pursuing graduation in Computer Science and Engineering from Kalasalingam Academy of Research and Education of batch 2017-2021. She had completed her schooling in Sri Chaitanya School, Hyderabad, Telangana. She had a membership in Association for Computing Machinery (ACM) for the student chapter of her institution.

Nagella Kedharnath, is currently pursuing graduation in Computer Science and Engineering from Kalasalingam Academy of Research and Education of batch 2017-2021. He had completed his schooling in Sri Vignan E. M School, Andhra Pradesh. He had a membership in Association for Computing Machinery (ACM) for the student chapter of his institution.

S. Vidya, is working as an associate professor in Computer Science Department from Kalasalingam Academy of Research and Education. Her area of interest is in wind speed forecasting in renewable energy.