Forecasting Bugs from the Software using Machine Learning Approach

Mr. Anmol Chhabra
Research Student, Dept. of CSE, Swami Vivekananda Institute of Engineering and Technology - Affiliated with Maharaja Ranjit Singh Punjab Technical University, Punjab, India,

Ms. Manmeen Brar
Assistant Professor, Dept. of Computer Science, Swami Vivekananda Institute of Engineering & Technology - Affiliated with Maharaja Ranjit Singh Punjab Technical University, Punjab, India.

Abstract: Machine Learning spreads the roots all over the globe and software engineering field is even not exempted from it. Forecasting the software bugs at the beginning phases of the software improves the efficiency, quality, reliability and many other aspects. The designing of software is highly complex which thus needs various guidelines [1]. So, machine learning algorithms use the supervised technique for model building and forecasting the bugs based on the historical data statistics. It thus reduces the development cost of the software. Three major algorithms are used as classifiers - Logistic regression, Decision Tree and Random Forest Classifiers. Hence it results in the model efficiency.

Index Terms - NFRs; IDE; UML; MSE; MMRE.

I. INTRODUCTION

There are two categories in the software such as system software and the application software. All the hardware components are managed by the system software so it can utilize as a functional unit. For the accomplishment of the specified task application software has been utilized. In order to ensure that the quality of the software is maintained as per the requirement, the software quality factors are to be computed. If the errors present within the software can be easily corrected and the new functions are easily added to the product and functionalities of the product, then the software product can be said to be maintainable [2].

Software Engineering is the discipline of computer science which applies engineering principles to create, operate, modify and maintain software components [3]. It is used to analyze software behaviour. The reverse engineering is the technology of software engineering in which source code is generated from the developed model. The abstract-present is the model of software engineering which is used to generate the source code from the sequence model. The code that is generated for the one phase will be given as input to generate code for the second phase. To generate reliable code, the improvement will be proposed in the abstract-present model. To do so, the SVM classifier will be used to classify required and non-required code to generate next phase of code. The potential NFRs present within the system are identified within the initial phase of this analysis.

The process that includes the design, manufacturing, assembling and maintenance of products and systems is known as engineering. The complete engineering process is classified into two parts which are forward engineering and reverse engineering. The process in which the process goes from high-level abstractions and logical designs to physical implementation of the system is known as forward engineering technique [4]. Requirements which define that the product that is delivered must behave in a particular way, e.g. execution speed, reliability etc [5].

II. PROBLEM FORMULATION

The reverse engineering is the process which can generate the code from the developed UML model. To derive code efficiently with reverse engineering, the sequence diagram plays an important role for the model-generation. In the base paper, abstract-present model is used to implement reverse engineering. In the abstract-present model, the source code of one stage will be analyzed to develop the source code for the second stage. In this model, the abstract-syntax tree is derived for the generation of syntax code. In the base paper, the sequence diagrams are visualized to derive new phase source code from the sequence diagram. In this research work, the technique of classification will be used to identify required source code which increases the reliability of the developed model.

Following are the various research gaps which will be covered in this research work:-

1. The sequence diagram defined the whole flow of the project which needs to classify. In the base paper, the required and non-required functions are identified on the basis of sequence diagrams. In this research work, dataset of the function will be collected for the classification based on SVM classifier.

2. The execution time of the base paper algorithm to classify functional and non-functional requirement functions are too high which need to be reduced.

III. TOOL USED

The Tool used for the analysis of datasets using Machine Learning Algorithms includes Anaconda and Spyder. The language used is Python.

Python is object-oriented, a high-level programming language with dynamic semantics. It is the general-purpose interpreted language. It is used for Rapid/Quick Development. It is simple and easy to learn the language. Its edit-test-debug cycle makes this language very effective.

Python supports various packages and the modules which make this language modular and the feature of code-reusability can be achieved. Also, it is an open-source language, so it can be used without any cost.

Libraries to be used in the Project –

a) Numpy: This package is used to compute various numerical and scientific calculations. Its functions include linear algebra, Fourier transformations and random number calculations.

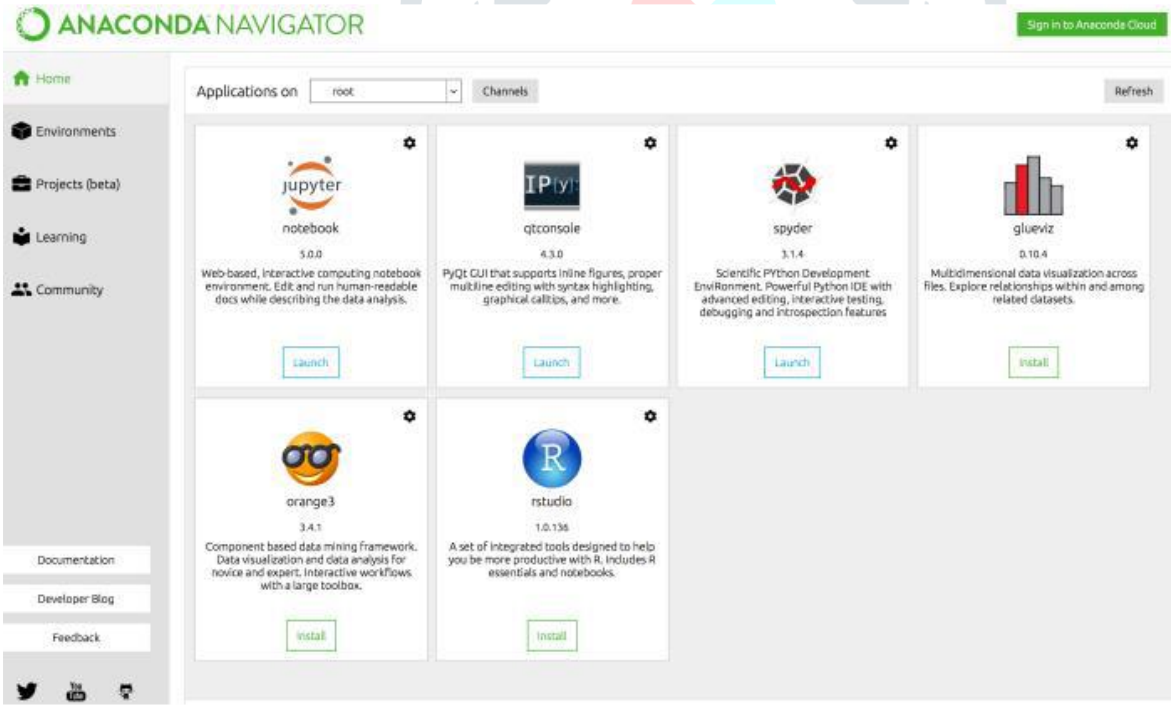
b) Mathplotlib: Collection of various command style functions which make this library to act as MATLAB. Its functions include plot areas in the figure, plot some lines in that figure, decorate plots with different labels, etc.

c) Pandas: This library is used for data analysis and manipulation. Data can be analyzed using Series (1-D array used to store any data type) and DataFrames (2-D array used to store data in rows and columns).

IV. RESULTS AND DISCUSSION

4.1 *Default Interface of Anaconda*

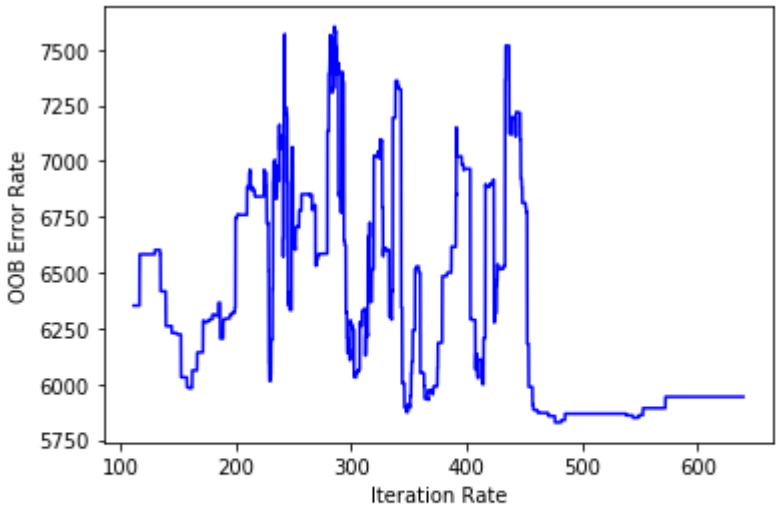
Anaconda Navigator is a desktop GUI that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. It is available for Windows, macOS, and Linux.



4.2 Comparison of Existing and Proposed Model

4.2.1 NFS Implementation (Existing Method)

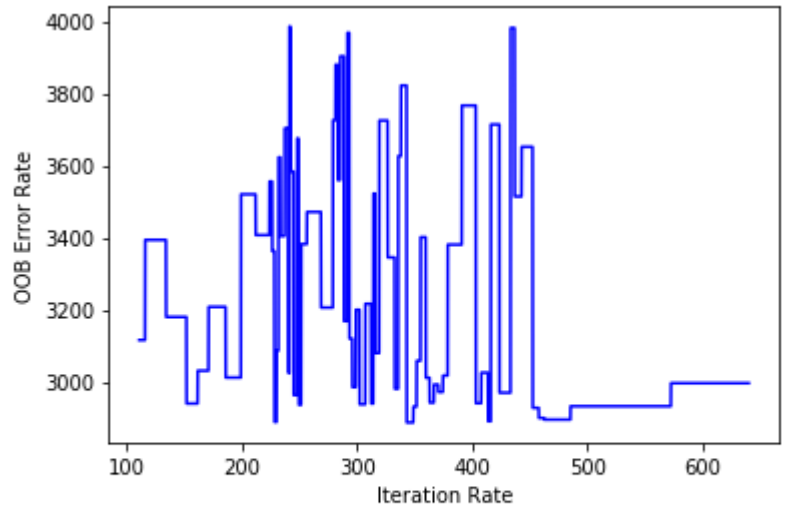
In this method, the accuracy and prediction factors are computed using the Random Forest Regression Algorithm. In graph, MSE is Mean Squared Error i.e. the averaged square difference between the estimated value and true value. MMRE represents Mean Magnitude Relative Error to calculate the mean of relative errors.




```
The accuracy is = 74.49
The MSE is = 58.0
The MMRE is = [ 0.377004]
The PRED is = 63.518728
```

4.2.2 Proposed Methodology

The proposed methodology uses Decision Tree Algorithm and computes the accuracy and prediction.




```
The accuracy is = 79.62
The MSE is = 57.75
The MMRE is = [ 0.354234]
The PRED is = 79.7
```

V. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

It has been concluded that reverse engineering is an efficient technique to generate source code. The process of extraction, abstraction, and visualization is used for the generation of source code. The classification technique used is Decision Tree Classification which provides the more accurate results and having less error rates.

5.2 Future Scope

Software defect prediction using machine learning algorithms is a very efficient and effective way to predict the defects in any software application in very less time. Further investigation can be done by different machine learning algorithm for improvement in the prediction accuracy.

VI. ACKNOWLEDGEMENT

I am thankful to the creator of this universe who created me as a human being. I am also thankful to my guide and my family members, who always support, help and guided me during my work. I specially thank to my childhood teachers who groomed me well to innovate new ideas in this world of competition. I especially thanks to my special to the unparallel and unmatchable mother who is there like a sheath and shadow for me. She always supports my innovative ideas. I am thankful to my college friends who helped me a lot and at last, I am thankful to all relatives who pray for my astonishing success.

REFERENCES

[1] C. Willems and F. C. Freiling, "Reverse code engineering - state of the art and countermeasures," it - Information Technology, vol. 54, no. 2, pp. 53–63, 2012.
[2] P. C. V. Oorschot, "Revisiting Software Protection," in ISC 2003. LNCS. Springer, 2003, pp. 1–13.
[3] S. E. Quadir et al., "A survey on chip to system reverse engineering," JETC, vol. 13, no. 1, pp. 6:1–6:34, 2016.
[4] U. Guin et al., "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," Proceedings of the IEEE, vol. 102, no. 8, pp. 1207–1228, 2014.
[5] F. Shull, et al., How Perspective-Based Reading Can Improve Requirements Inspections, IEEE Computer, Vol. 33, No. 7, pp. 73-79, Jul.2000.