Major Project Synopsis

On

Topic

Nature-inspired Approaches in Software Faults Prediction

Mentored By -

Mr. Varun Goel & Mr. Sachin Garg

Student Details -

Student Name: Harshit Saini, Tushar Arora

Roll No: 20214803119, 20514803119

Semester: 8th Semester

Group: 8-I-8

Maharaja Agrasen Institute of Technology, PSP Area, Plot No. 1, Sector – 22, Rohini, New Delhi – 110086

TITLE OF THE PROJECT

”Nature-inspired Approaches in Software Faults Prediction”

To develop or find algorithms with the help of machine learning using nature-inspired optimization to predict faults in software, which thereby increases software quality and testing efficiency.

PROBLEM STATEMENT

During recent years, we have witnessed a trend in which most of modern needs have become dependent on software. We can find it in our toothbrushes, watches, cars, etc. The IT and software industry has grown tremendously over the past few years, creating an increasing impact on the lives of people and on society as a whole. Consequently, we must make the software and applications more accurate, free of major errors, and more reliable. The more reliable the software is, the more valuable it becomes, and the less likely the software is to contain faults. Therefore, predicting software flaws could be very useful in the IT field and will have a profound impact on society at large.

Why is the particular topic chosen?

As stated in the problem statement, predictive fault detection is very important if we are going to improve software reliability and have a positive impact on the IT sector. Having been studying the IT field ourselves, we cannot miss the fact that we run into problems when using different software. It will be a great way to contribute to society as well as we will be learning a lot while developing and researching different solutions for the problem.

OBJECTIVE & SCOPE

The main objectives of this project are as follows:

- To determine algorithms using machine learning that can perform better in predicting the faults in software than the existing ones.

- To improve the results with the help of current datasets available on the internet using our algorithms.
- To use different evaluation benchmarks to evaluate the performance of our model.
- In extension to our minor project, we will use nature-inspired optimization to further optimize the results obtained from different attributes of the datasets used.

The overall scope of the project is to create an automated software fault prediction model using machine learning techniques that will make the software more efficient and reliable to use. The main task of this project is to use nature-inspired optimization algorithms on the datasets to get more refined results.

METHODOLOGY

Software defect detection is mainly used for identifying the defective modules that are present in the software so that it can help in improving the quality of the software system. Machine learning techniques are useful in terms of software defect detection. Software is divided into defective and non-defective modules. For the classification of defective data sets, decision tree and logistic regression algorithms are implemented. The datasets are taken from the promised data repository and, thereby, accuracy is computed. The algorithms are implemented using the Jupyter Notebook as well as the Python IDE, and the comparative analyses of the results are displayed.

RESOURCES

Software used will be Jupyter notebook, Datasets using Kaggle, Python IDE, Machine Learning algorithms with Nature Inspired Optimization

Hardware will not be used during this project.

CONCLUSION

What contribution would the project make towards the society?

We aggregated those parameters and analysed them accordingly, and we also illustrate the different challenges in the SFP domain. We also compared the performances of machine learning and statistical techniques based on SFP models. Our empirical study and analysis demonstrate that the prediction ability of machine learning techniques for classifying class/module as fault/ non-fault prone is better than classical statistical models. The performance of machine learning-based SFP methods over fault susceptibility is better than conventional statistical purposes. The empirical evidence of our survey reports that the machine learning techniques have the capability to identify fault proneness and are able to form well-generalized results.

SCHEDULE OF THE PROJECT

March 2023

We will begin by analyzing Kaggle data sets. Furthermore, research papers will be reviewed as well as previous algorithms and results.

t

April 2023

Then, we will test different algorithms that we didn't use in the minor project and will also use nature-inspired optimization to determine which one produces the best result, and we will try to improve the results.

May 2023

Ensure that algorithms are optimized by consulting mentors and reviewing changes in response to feedback. In addition, we will test the model against different available datasets for web, desktop and mobile based softwares.

REFERENCES & BIBLIOGRAPHY

[1] The Institute of Electrical and Electronics Engineers. “29119 -1-2013 - Software and systems engineering —Software testing.” IEEE Standards Association.

[2] International Organization for Standardization. “What Is a Standard?” ISO Standards.

[3] Naik K, Tripathy P. Software Testing and Quality Assurance. Theory and Practice. Wiley, 2008, 616p.

[4] Myers G, Badgett T, Sandler C. The Art of Software Testing. 3rd Edition. Hoboken, NJ: J. Wiley & Sons; 2014.

[5] Spillner A, Linz T, Schaefer H. Software Testing Foundations. 4th Edition.. Santa Barbara: Rocky Nook Inc.; 2014.

[6] Meyer B. Seven Principles of Software Testing. Computer 2008, August:99-101.

[7] Autili M, Di Salle A, Gallo F, Perucci A, and Tivoli M., Biological Immunity and Software Resilience: Two Faces of the Same Coin?, in A. Fantechi and P. Patrizio (Eds.): SERENE 2015, LNCS 9274, DOI:10.1007/978-3-319-23129-71, 1–15, 2015.

[8] Madsen H, Thyregod P, Burtschy B, Albeanu G, Popentiu F. A Fuzzy Logic Approach to Software Testing and Debugging. In: Guedes Soares, Zio E, editors. Safety and Reliability for Managing Risk , London:Taylor & Francisc Group; 2006, p. 1435-1442.

[9] Madsen H, Thyregod P, Burtschy B, Albeanu G, Popentiu F. On using soft computing techniques in software reliability engineering. International Journal of Reliability, Quality, and Safety Engineering 2006;13(1):1–12.

[10] Sharma C, Sabharwal S, Sibal R. A Survey on Software Testing Techniques Using Genetic Algorithm. International Journal of Computer Science Issues 2013;10(1):381-393.