

Machine Learning Approach for Software Defect Prediction

Shivani Sharma¹, Er. Poonam Chaudhary²

¹Mtech Scholar, ²Assistant Professor

^{1,2}Sirda Institute of Engineering Technology, Sunder Nagar

Abstract- The software engineering is the technology which is used to analyze software behavior. 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.

Keywords- SVM, Back tracking, Software Defect Prediction

I. INTRODUCTION

A collection of programs, procedures, data and documentation is known as software. By keeping in consideration the hardware and operating system which is also known as a platform, the software is designed for a particular organization. A systematic method through which the software is developed is known as engineering. The designing of software is highly complex which thus needs various guidelines [1]. 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. A software product is known as when it is made for a specific requirement as it contains many utilities and operating system for the purpose of disk formatting, file managers, display managers, etc. For the accomplishment of the specified task application software has been utilized. Software Engineering is the discipline of computer science which applies engineering principles to create, operate, modify and maintain of software components [2]. Software engineering is about building, evolving and maintaining software systems. Software engineering is a set of problem solving skills, techniques, technology and methods applied upon a variety of domains to evolve and create useful systems that solve many problems like practical problems. Software engineer is required to handle software engineering projects which discover, create, build software and tells its behavior. An organized and systematically approach is adopted by software engineers regarding their work using some techniques and tools depending upon the resources available and problem to be solved [3]. System engineering is different from software engineering. System engineering is concern with deployment, architectural design and integration where as software engineering is concern with development,

quality and testing and control of the system. 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]. There might be a physical product included here which might have no technical details in some scenarios. The method in which there are no drawings or documentation or computing model included for duplicating the existing part is known as reverse engineering process. The process in this the geometric CAD model is achieved from 3-D points that are gathered from the scans of existing parts can also be defined as reverse engineering process. The physical entities of a component can be digitally captured here on the basis of a particular task to be completed. In the software development activity the first step is to determine and document the system-level requirements. This can be done in system requirements specification or in software requirements specification or in both [5]. There are some responsibilities that must be fulfilled in the software requirements such as a problem must be solved or achieve an objective as well as capabilities that a system or component needs to satisfy a contract, standard, or other formally imposed document. After the announcing of the acquirer and user system requirements this process of analysis in software requirements starts [6]. The identification of all software system requirements and its conclusion are the included functions that establish baseline requirements known as the allocated baseline. In the event that an organization wishes to give an agreement for a substantial programming improvement a chance to extend it must characterize its needs in an adequately dynamic manner that an answer isn't predefined. The prerequisites must be composed with the goal that few temporary workers can offer for the agreement, offering, maybe, unique methods for addressing the customer association's needs. The needs of customers for a system are reflected by the Functional requirements. The planned behavior of the system is captures using this requirement. The system is required to perform all the behavior may be expressed as services, tasks or functions [7]. It provides all the details, that how system should provide the statements of services and to the particular inputs how this system should react. Also provide the details that how system should behave in particular situations. An essential role is

played by the Non-functional requirements in the software and systems engineering. In the field of NFRs, there is much work to do such as classifications and characterization of single classes of NFRs. It also provides the security and reusability. It is considered as the most important activity in requirements engineering. The system requirements are the description of the provided services by the system and its operational constraints.

II. LITERATURE REVIEW

Senay Tuna Demirel, et.al (2018) presented in the software development projects a major key challenge role is played by the requirement analysis. In the software projects, the involvement of the specific customer requirements and management has various impacts. Hence, it is required to improve this area more in terms of both academic and industrial fields. They presented the model of CMMI in order to expose the development and management requirement. It also specifies the various different goals and practical platform for them as well [8]. They listed in this paper the requirement of the management and key challenges and issues faced by it with the help of the model CMMI and its regular activity.

Syed Waqas Ali, et.al (2018) presented the technical documentation plays an essential role in determining the success and failure of any software for which a Software requirement specification document has been utilized. Hence, to resolve this author in this paper proposed an effective approach [9]. In this method, four processes were involved such as Parsing Requirement (PR), Requirement Mapping using Matrix (RMM), Addition of Requirements in SRS template and Third Party Inspection. To the PR, required input was provided by the process of requirement engineering after which it is implemented on its ontology rules due to which it fulfill the requirements. A detailed report to team of requirement engineers was submitted by the third party after performing detailed analysis using inspection models and assigning total quality score.

Ra'ul Navarro-Almanza, et.al (2017) proposed an automation of SRC with the help of the natural language processing and information retrieval techniques. This technique requires human effort so that created features can be analyzed easily [10]. They proposed the use of deep learning in order to classify the requirements of the software without utilizing the feature engineering. On the basis of the Convolutional Neural Network this proposed method also stated as art in other languages. They utilize the PROMISE corpus for the evaluation of the proposed method in which a set of labeled requirements in functional and different categories of nonfunctional requirements was utilized. On the basis of the performed experiments, it is concluded that

proposed method shows effective results for the SRC using CNN.

Jorge Rafael Aguilar Cisneros, et.al (2017) presented an essential role is played by the safety system in the automotive sector. These safety systems are ABS brake systems, stability control system, and air bag systems. Among these safeties system one is oriented towards collisions which are activated at different interval of time with objectives [11]. The objective of minimizing the physical harm to occupants these systems is activated during collision. As per done survey, it is determined that type 1(passive) and type 2 (active) has been utilized widely while types 3 are under research. The software requirement specification document represented the behavioral logic of a type 3 system. Some of the recommendations of the requirements are satisfied by the presented SRS in the automotive standard ISO 26262 and to the requirements of the standard ISO/IEC/IEEE 15288 they utilized the processes in this paper.

Paula Fraga-Lamas, et.al (2017) presented in this paper that within various applications such as in public transportation systems, the Radio Frequency Identification (RFID) security is not handled in appropriate manner. In order to reverse engineer and identify the security flaws, a novel methodology is proposed in this paper. There is an analysis presented here of the various communications protocols being utilized in ISO/IEC 14443-B public transportation card by numerous users. The private information can be accessed by the application of methodology with the hardware tool. The tag-reader communications can be captured and the tags and readers are emulated within this proposed methodology as well [12].

Tuan Anh Nguyen, et.al (2017) presented in this paper that the automatic pinpointing of the classes of object-oriented program which implement the interesting domain concepts would be very helpful for maintaining the industrial software. There are two observations encoded by the programmer behavior within Reoom. Reoom is a novel light-weight static analysis technique utilized within such methods. Various comparisons are made amongst Womble and the various other third-party open source applications. It is seen through the various simulation results that the results show better achievement in terms of overall precision, recall as well as accuracy. There is a good tradeoff amongst full Reoom in comparison to Womble. There is a better recall provided by the locals and parameters here, in comparison to the return types [13].

III. RESEARCH METHODOLOGY

The potential NFRs present within the system are identified within the initial phase of this analysis. On the basis of major assumptions of the cluster hypothesis, the research

assumptions are generated within this phase. On the basis of relevance of information required, the documents that behave similarly are present within similar clusters. On the individual words of information content present in the documents, this behavior is attached. Thus, the grouping of words within similar cluster is done specifically. On the basis of this approach, certain assumptions are used. The generation of semantically coherent groups of natural language words is the major objective of clustering the keywords that are extracted from the functional requirements of software system. Group of words $W = \{w_1, w_2, \dots, w_m\}$ into a set of disjoint groups of semantically similar words are clustered such that a $C = \{ [c_1, c_2, \dots, c_k] \}$ is generated which is a set of disjoint groups that include semantically similar words. The conceptual themes that pass through the original text need to be described ideally by the clusters in C. the quality constraints of a system are mainly represented with the help of such themes. A NP-hard issue that arises here is the identification of optimal cluster configurations that are best fitted. However, with the help of certain experiments, the near-optimal solutions can be achieved. The performances of series of semantic similarity measures as well as the word clustering techniques are evaluated in order to determine such configurations. Thus, from the FR words, the semantically coherent or thematic clusters can be generated.

A feature of an instance is represented by each node in a decision tree for the classification assumed value by the node is represented by each branch. These instances are classified at the root node initially after which on their feature values they are sorted. This learning has been utilized in data mining and machine learning. It is a predictive model in which item is observed on the map in order to identify item's target value. These tree models are also known as classification trees or regression trees. For the evaluation of decision trees, post-pruning techniques has been utilized as they are pruned by using a validation set. This research work is based on the classification of null and non null function from the IVR dataset. To do so, the NFR matrix is applied with the decision tree classifier for the data classification. The performance of proposed and existing method is compared in terms of accuracy and execution time. In the existing method only NFR matrix is applied for the data classification. In the proposed method, the decision tree is applied with NFR matrix for the classification.

IV. EXPERIMENTAL RESULTS

The proposed research is implemented in Python and results are evaluated in terms of various performance parameters.

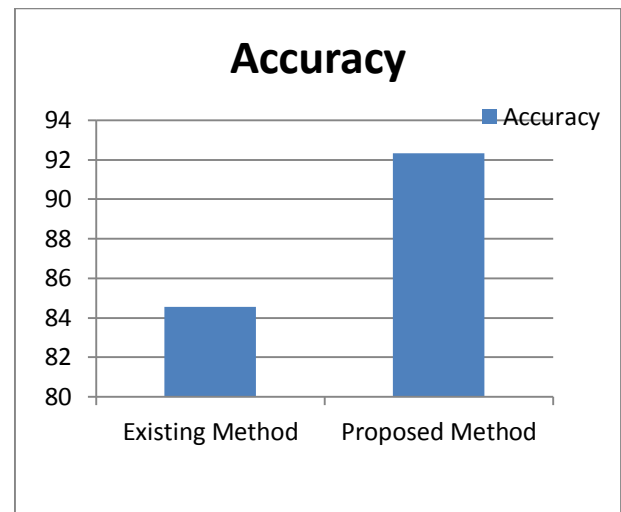


Fig.1: Accuracy Comparison

As shown in figure 1, the accuracy of proposed method is compared with the existing method. It is analyzed that accuracy of proposed method is high as compared to existing due to use of decision tree with NFR matrix

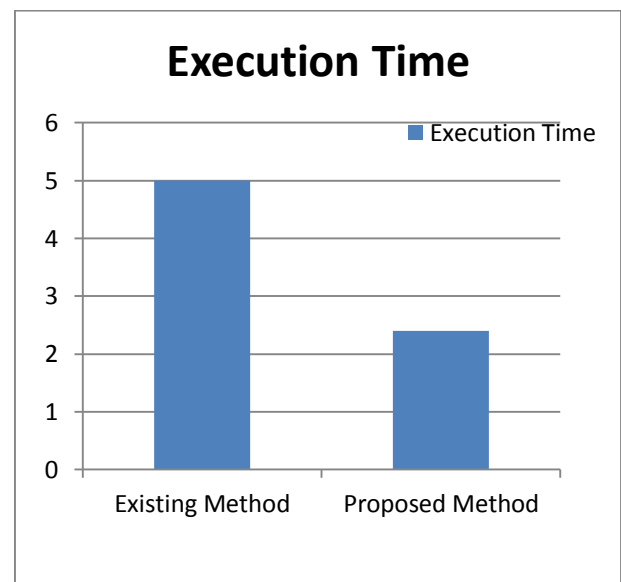


Fig.2: Execution Time

As shown in figure 2, the execution time of proposed method is less as compared to existing method. It is due to use of NFR matrix with decision tree for the classification

V. CONCLUSION

In this research work, it has been concluded that reverse engineering is the efficient technique to generate source code. In the base paper, the abstract-present model is used to generate the source code. In the abstract-present model, the

code of previous sequence is used to generate code of new sequence. The process of extraction, abstraction and visualization is used for the generation of source code. In this research work, the classification technique will be used before visualization which increase the reliability of abstract-present model of reverse engineering.

VI. REFERENCES

- [1]. Dang Viet Dezung, Atsushi Ohnishi Improvement of Quality of Software Requirements with Requirements Ontology "International Conference on Quality Software , Jeju, pp 284-289, 2009.
- [2]. F. Shull, et al., How Perspective-Based Reading Can Improve Requirements Inspections, IEEE Computer, Vol. 33, No. 7, pp. 73-79, Jul.2000.
- [3]. C. Denger and T. Olsson, Quality Assurance in Requirements Engineering, in Engineering and Managing Software Requirements, A. Aurum and C. Wohlin, Eds. Springer, pp. 163-185, 2005
- [4]. J. G. Hall, M. Jackson, R. C. Laney, B. Nuseibeh, and L. Rapanotti, "Relating software requirements and architectures using problem frames," in Proc.of the IEEE Joint International Conference on Requirements Engineering, 2002, pp. 137-144.
- [5]. P. Grnbacher, A. Egyed, and N. Medvidovic, "Reconciling software requirements and architectures: The cbsp approach," in Fifth IEEE International Symposium on Requirements Engineering, 2001, pp. 202-211.
- [6]. M. Brandozzi and D. Perry, "Transforming goal oriented requirement specifications into architectural prescriptions," in Proceedings of First International Workshop From Software Requirements to Architectures (STRAW01), 2001, pp. 54-61.
- [7]. D. Liu, K. Subramaniam, A. Eberlein, and B. H. Far, "Natural language requirements analysis and class model generation using UCDA," in Lecture Notes in Computer Science. Springer-Verlag, 2004, pp. 295-304.
- [8]. Senay Tuna Demirel, Resul Das, "Software Requirement Analysis: Research Challenges and Technical Approaches", IEEE, 2018
- [9]. Syed Waqas Ali, Qazi Arbab Ahmed, Imran Shafi, "Process to Enhance the Quality of Software Requirement Specification Document", IEEE, 2018
- [10].Raúl Navarro-Almanza, Reyes Juárez-Ramírez, Guillermo Licea, "Towards supporting Software Engineering using Deep Learning: A case of Software Requirements Classification", 2017 5th International Conference in Software Engineering Research and Innovation
- [11].Jorge Rafael Aguilar Cisneros, Genaro de la Rosa Garcia, Carlos Alberto Fernández-y-Fernández, "Software Requirement Specification for the Automotive Sector: The case of a Post-Collision Event Control System", 2017 5th International Conference in Software Engineering Research and Innovation
- [12].Paula Fraga-Lamas, Tiago M. Fernandez-Carames, "Reverse Engineering the Communications Protocol of an RFID Public Transportation Card", 2017 IEEE International Conference on RFID (RFID)
- [13].Tuan Anh Nguyen, Christoph Csallner, "Reverse Engineering Object-Oriented Applications Into High-Level Domain Models With Reoom", 2017 IEEE/ACM 39th IEEE International Conference on Software Engineering Companion