International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Software Defect Prediction Mechanisms for Quality Enhancement

Amutha A, V ArulKumar , Francis Densil Raj V ,

Abstract: ***Software engineering is an imperative field that helps to provide a variety of software to the different industrial domains. In this regard, providing quality software is a challenging task. Now a lot of advancements are occurring in this field to enhance the software quality. But still software failures will occur at the development stage. To overcome these difficulties, various failure detection mechanisms were adopted. Among them, software defect prediction is one such mechanism to provide quality software. The efficacious of testing the software and guiding the distribution of resources can effectively enhanced by Software defect prediction (SDP). This research article will provide a complete knowledge about the defect prediction mechanism in the software domain. And also profound analyses were done with the existing literature to know how this mechanism will ameliorate the quality of the software product.***

Index Terms: Software Defects, Defect Prediction, Software Quality, Machine Learning Algorithm.

I. INTRODUCTION

In the software development process, failure can occur and it preeminent to software defects. Software defect is a fault that is acquainted by stakeholders and software developers. Software defect is also pertained as a bug that causes the software not to accomplish its task as the developer and customer needed. Defects can happen in any stage of software development. Software defects are expensive if ascertained and fixed in the subsequent stages of the testing and development life cycles. Developing software companies mainly concentrate on software quality. SDP is a successful research region in programming designing to pinpoint imperfection inclined modules [1]. The principle target of SDP is to produce more trustworthy software by reducing development time, cost, and rework effort. Models which equipped for recognizing where flaws are in code have the capacity to spare organizations a lot of cash. Defect prediction is a very pivotal and indispensable activity that helps the software developers to effectively improve the apportion of limited resources for testing and maintenance.

Fig.1 represents that software defect prediction can identify defective modules and non defective modules by using a variety of software measures and historical data**.**

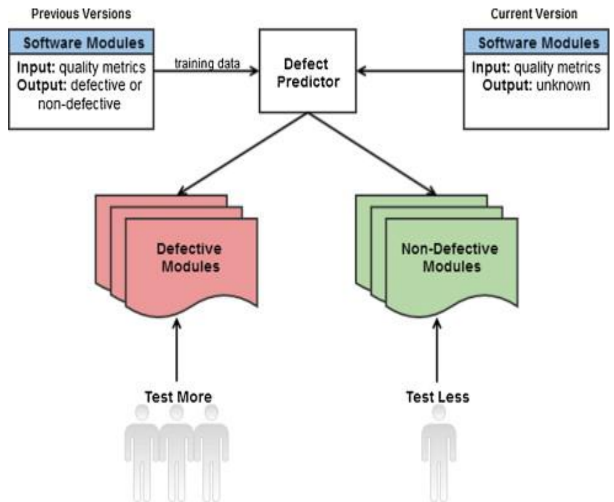

Fig.1 Software defect prediction [7].

In this research paper, we give a thorough investigation of significant methodologies, algorithms, models and research hotspots that can be utilized for the performance of SDP. Furthermore, we endeavored to evaluate the exploration works as for programming defects. This article examines the state of skill in the field of SDP. Section 2 deals with literature survey. Section 3 explains the Software Defect Prediction approaches. Section 4 presents the process of a SDP model and lists some machine learning algorithms. Section 5 lists the evaluation measures. Software defect prediction models used in the research are explained in Section 6.Section 7 presents some of the results of the evaluation experiments conducted on the software defect prediction models. Section 8 concludes this work.

II. LITERATURE REVIEW

In this Literature survey, various research articles related to software defect prediction model are reviewed and are represented in the following.

Lech Madeyski, et. al. [1], proposed an empirical evaluation on process metrics and product metrics to enrich the defect models. For this, four different product metrics and process metrics were broke down so as to determine their handiness in the SDP. Finally they concluded that process metrics highly contribute to defect prediction model than the product metrics.

Manuscript published on 30 September 2019
* Correspondence Author
Amutha A*, School of Computer Science and Applications, REVA University, Bangalore, India.
Dr V ArulKumar , School of Computer Science and Applications, REVA University, Bangalore, India.
Francis Densil Raj V , School of Computer Science and Applications, REVA University, Bangalore, India.

In product metric, security is one of the key metric which is handled by Amutha et. al., [2] by proposing the Symmetric key Ciphers.

Duksan Ryu, et.al. [3], examined k-nearest neighbor and Naive Bayes-NB .To identify the defect prediction rate, they used the probability of detection and false alarm. They asserted that a hybrid instance selection using nearest neighbor (HISNN) is better than classification models and yield high overall performance to manage software quality.

Pradeep Kumar Singh, et. al. [4], has discussed about defect prediction to identify the susceptibility in the project. The author has studied various methods to evaluate the software defects. After analyzing various papers, the author recognized the reasons influencing the imperfection that can be helpful to predict the bugs.

Suneetha Merugula R [5], identified the defect which aids to improve testing. The classification techniques examined are statistical techniques, clustering techniques, Rule-Based Techniques, Artificial Neural Networks (ANN), Nearest Neighbor, Support Vector Machines (SVM), Decision Trees (DT), Ensemble Methods .They categorized into two projects of within project defect prediction (WPDP) and Cross project defect prediction (CPDP) to reduce the effort in locating defects. Finally, the author asserted that SDP model using classification techniques helps in early detection of defects.

B. Dhanalaxmi et.al. [6], suggested that researchers should concentrate on experimental components like validation techniques, Noise procreated by issue report mislabeling and parameter settings to create more veracious and reliable defect prediction models. Finally, an author demonstrated that the experimental components can be utilized to assemble SDP models in terms of accuracy and reliability.

Tanvi Sethi , et.al.[7],provides the frame work for SDP using artificial neural network. Data set from twenty genuine software undertakings were used for the survey. The author compared the implementation of neural network based SDP and the results of the fuzzy logic basic approach. Finally, the study results showed that ANN based approach provides surpass the results with more accurate and precision based values for better performance.

Logan Perreault, et. al.[8], has successfully applied classifier models to the assignment of identifying defects in Software Engineering. These models assist to higher apprehend the underlying software program functions and how they have an effect on the defects. In this paper, they evaluated the behavior of five distinctive classifiers within the milieu of SDP. Results showed that all models can detect software defects, but different classifiers will produce different accuracy. On various analyses, they declared that NB and SVM machines give the best performance of the software defect prediction.

Hans Raukas [9], presented a generic description of SDP models with machine learning (ML) classifiers. They suggested a brief explanation of the algorithms, like Logistic Regression, NB, Random Forest (RF), K-Nearest Neighbor, SVM, K-Means Clustering, and Particle Swarm Optimization. They also discussed about evaluation parameters like recall, Precision and F-score to assess the prediction accuracy. After analyzing, they pointed out that the best defect prediction model for CPDP and WPDP based on the average F-score is Transfer Component Analysis Neural Network and P- Support Vector Machine respectively.

He, Peng, et .al. [10], provide an enhanced method by using training data, mentioned as TDSelector for CPDP. Moreover TDSelector, can provide their weights to achieve the first-rate result.. They conducted the analysis of various projects. The effects advise that the TDSelector method can be used to improve software program best.

N.Kalaivani, et.al.[11], Suggested various data mining techniques to improve the prediction models. They considered WPDP, CPDP for Similar Dataset, and CPDP for Heterogeneous Dataset. For this they examined supervised and unsupervised algorithms for the prediction models. The author concludes after assessing the previous articles related to software defect, a various techniques of data mining were used to predict the software faults.

Bowes,et.al. [12], examined that distinct classifiers come out with same defects. For this, four classifiers like NB, RF, RPart and SVM were investigated with different data sets. After analyzing, results confirm that each classifier detects different sets of defects.

Awni Hammouri, et. al,[13], discussed a SDP model using machine learning (ML) algorithms. Three supervised ML algorithms like NB, DT and ANN were accustomed. These classifiers were applied to 3 various data sets and comparison is done on the basis of different evaluation measures. The assessed results revealed that the DT approach in ML algorithms can provide superior performance over the others.

Zhiqiang Li,et.al.[14], aimed to predict defect-prone software modules. This model was used to provide better quality assured software. They summarize all typical defect prediction papers as of late. In light of the outcomes acquired, the challenges identified assist programming practioners to be aware about the defect prediction studies from datasets, software metrics, evaluation measures, and modeling techniques perspectives in a simple and successful manner.

III. SOFTWARE DEFECT PREDICTION APPROACHES

Three predominant methodologies performed to assess prediction models are as follows [11].

- WPDP- A prediction model is fabricated by gathering local information from a software product venture and foresee defects in a similar undertaking are known as WPDP.
- CPDP for Similar Dataset- A prediction model is constructed for one project and it is applicable to some other project.
- CPDP for Heterogeneous Dataset - Predict defects with imbalanced metric sets across projects.

IV. SOFTWARE DEFECT PREDICTION PROCESS

The figure 2 depicts the general defect prediction process. The steps involved in the process are as follows [15].

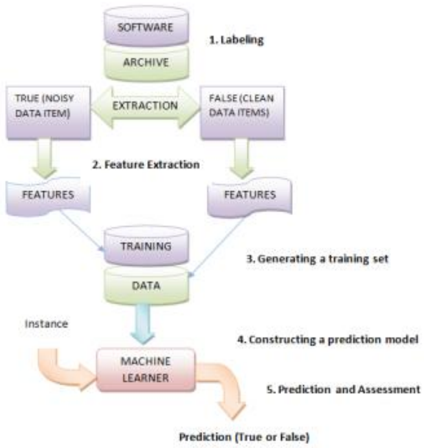

Fig. 2 General Defect Prediction Process [15]

4.1 Labeling:
The first step of the process is collecting files known as instances from software library and labeled as true (buggy) or false (CLEAN). The labeling task is done on the basis of post-release-defects for each instance. If there is one post release-defect, it is mentioned as bug
gy. Else, it is labeled as clean.

4.2 Feature Extraction and inducing training sets:
This step entails excerpting the features like keywords, complexity metrics, and structural dependencies. By joining labels and features of files, the training set produced is used by a machine learner to build a prediction model.

4.3 Building prediction models:

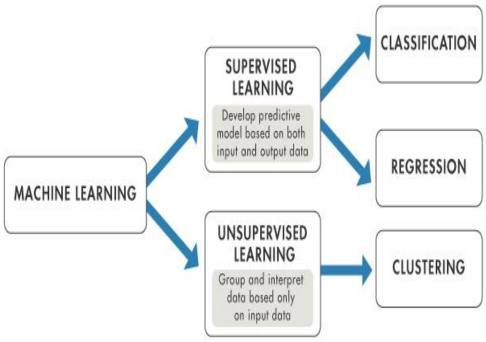

Fig:3 Machine Learning Algorithms [11].

Fig.3 shows the algorithms used in the SDP models [10]. In this nabe they are explained beneath.

4.3.1 Supervised Learning
In this learning, a model is able to predict with the help of already known trained dataset. Classification problem is where we can categorise the output variable. Regression method is concerned with building predictive model for function with real values. Classification and regression methods are enumerated in table: 1

Table: 1 Types of Supervised algorithms in Machine learning are tabulated.

SUPERVISED ALGORITHMS	
CLASSIFICATION TECHNIQUES	REGRESSION TECHNIQUES
•DECISION TREE •SUPPORT VECTOR MACHINE (SVM) •K-NEAREST NEIGHBORS •NAIVE BAYES •RANDOM FOREST •NEURAL NETWORKS	• LINEAR REGRESSION • LOGISTIC EGRESSION • POLYNOMIAL REGRESSION • LASSO REGRESSION • MULTIVARIATE REGRESSION

4.3.2 Unsupervised Learning
In this learning, there is no preceding information and machine tries to identify pattern and give their response. There is no training data.
Most significant unsupervised learning is clustering, which will make diverse group of inputs and will probably put new input to a suitable group. Clustering algorithms go under this category are tabulated in table 2.

Table: 2 Types of Unsupervised algorithms in Machine Learning.

UNSUPERVISED ALGORITHMS
CLUSTERING ALGORITHMS
K – MEANS CLUSTERING HIERARCHICAL CLUSTERING MAKE DENSITY BASED CLUSTERING

Training set can be used to construct the prediction model with machine learners. The model then can acquire a new instance and predict its label as TRUE or FALSE.

4.4 Assessment
To assess a prediction model, we require testing data set behinds a training data set. The data used to construct models comprise a training set, whereas those used to test the learned models comprise a test data. The names of files in the testing set are anticipated and the prediction model is broke down by comparing the prediction and real labels. In WPDP, both training and test sets are from the identical project. For CPDP, the training set and the test set are from different projects.

V. EVALUATION MEASURES

Evaluation parameters are utilized to assess the precision of a prediction model. It can be built by utilizing both classification and clustering algorithms. Individual effective measures are available for both classification and clustering techniques. Some of the Evaluation measures for Classification and Clustering are tabulated in Table: 3.

Retrieval Number: C4741098319/19©BEIESP
DOI:10.35940/ijrte.C4741.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table: 3 Classification and Clustering evaluation measures

CLASSIFICATION	CLUSTERING
• RECALL • PRECISION • F-MEASURE • ACCURACY • G-MEASURE	• TIME TAKEN • CLUSTER INSTANCE • NUMBER OF ITERATIONS

Precisely anticipating measuring units enables experts to target flawed units and organize their endeavors to keep up programming quality.

VI. DEFECT PREDICTION MODEL

TCA+

TCA+ is an improved rendition of the transfer learning method TCA (Transfer Component Analysis) which is used for cross-project software defect prediction.

Peters Filter

Peters filter is a CPDP approach that creates a filtered set of training data for a defect predictor to train on which then analyses the instances of a target project to be defective or clean.

ExtRF

ExtRF is a WPDP method that is an elongation of the Random Forest (RF) approach.

TCANN

TCANN (Transfer Component Analysis Neural Network) is cross-project software defect prediction approach that combines methods for dealing with class imbalance to reduce noise reduction in data, transfer learning between source and target datasets.

P-SVM

P-SVM is a WPDP model using Particle Swarm Optimization (PSO) and SVM algorithms.

VII. PREDICTION RESULT ANALYSIS

The SDP models shrouded in this work have been evaluated with the common measure F-score.

Table:4 Average F-score result of evaluation experiments investigated on TCA+, HYDRA, Peters Filter, TCANN, extRF and P-SVM[9].

MODEL (CLASSIFIER)	DATASET	WITHIN-PROJECT / CROSS-PROJECT	AVERAGE F-SCORE
TCA+ (LR)	RELINK	CROSS-PROJECT	0.61
TCA+ (LR)	AEEEM	CROSS-PROJECT	0.41
TCA+ (LR)	PROMISE	CROSS-PROJECT	0.43
HYDRA (LR)	PROMISE	CROSS-PROJECT	0.54
PETERS FILTER (LR)	PROMISE	CROSS-PROJECT	0.40
TCANN (ANN)	RELINK	CROSS –PROJECT	0.66
TCANN (ANN)	AEEEM	CROSS –PROJECT	0.41
TCANN (ANN)	RELINK	WITHIN-PROJECT	0.71
TCANN (ANN)	AEEEM	WITHIN-PROJECT	0.54
EXTRF (EXTRF)	ECLIPSE SET	WITHIN-PROJECT	0.53
P-SVM (SVM)	JM1	WITHIN-PROJECT	0.82

The classifiers used within the models are Logistic Regression (LR), Artificial Neural Network (ANN) and Support Vector Machine (SVM). The datasets were conducted as follows:

- PROMISE refers to the dataset used in the experiments of HYDRA, TCA+ and Peters Filter.
- AEEEM and ReLink refer to the datasets used in the experiments of TCANN and TCA+ respectively
- Eclipse Set refers to the dataset used in the experiments of extRF
- JM1 refers to the dataset used in the experiments of P-SVM

Each row of Table 1 shows the average F-score result obtained from experiments conducted based on the combination of model, dataset and model training approach. The results obtained from the observation are

- TCANN is the best defect prediction model for CPDP with the average F-score .66 on the Relink dataset.
- Additionally HYDRA shows the next best result with the average F-score .54 and it also considered as the best cross-project defect prediction model out of the other cross-project models covered in this work.
- P-SVM is the best defect prediction model for within-project with the average F-score .82 on JM1 data set.
- The next best result is obtained by TCANN with the average F-score being 0.71 on the ReLink dataset.
  Figure:4 exhibits the average F-Score between classifier models for various datasets namely Relink, AEEEM, Promise, Eclipse set and JM1 for cross project and within project. It is observed that TCANN has maximum F-Score of 0.58 whereas Peter Filter has minimum F-Score of 0.40.

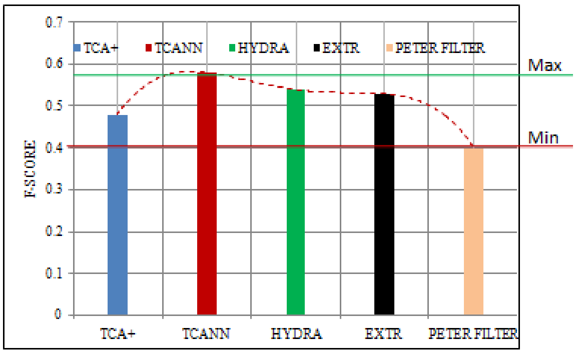

Fig: 4 Average F-Score for various classifiers

VIII. CONCLUSION

SDP is the method of identifying faulty components in software. The happening of defects is unavoidable, but we should try to circumscribe these defects to the lowest degree. One of the aspirations of this work was to give a general understanding of some of the processes of SDP models. In this regard, the general process of SDP using ML classifiers, a handful of specific models and a brief overview of the processes within those SDP models were presented. In addition, some of the results of the evaluation experiments conducted in the SDP were listed. These outcomes were dissected with the reason of providing the best defect prediction model for WPDP and CPDP.

REFERENCES

1. Lech Madeyski , Marian Jureczko .Which process metrics can significantly improve defect prediction models? An empirical study, Software Qual J, (2015): 23: pp .393–422.
2. Amutha A and H.Krishnaveni, A Study on Symmetric key Ciphers in Cryptography and Network Security, International Journal of Emerging Technology and Advanced Engineering, Vol. 8, no. 2, 2018, pp. 37-40
3. Duksan Ryu, Jong-In Jang and Jongmoon Baik, Member, ACM, IEEE. A Hybrid Instance Selection Using Nearest-Neighbor for Cross-Project Defect Prediction, Journal of computer science and technology 30(5): (2015). pp. 969–980 .
4. Pradeep Kumar Singh, Dishti Agarwal Aakriti Gupta ,A Systematic Review on Software Defect Prediction, 978-9-3805-4416-8/15. (2015) .
5. Suneetha Merugula, A Study on Software Defect Prediction Using Classification Techniques, International Journal of Computer Science & Engineering technology,(2016)Volume .7 N0.11 , ISSN:2229-3345.
6. B.Dhanalaxmi, Dr. G. Apparao Naidu, Dr.K.Anuradha,Practical Guidelines to Improve Defect Prediction Model – A Review, International Journal of Engineering Science Invention ISSN (Online): 2319 – 6734, Vol. 5 Issue 9, (2016) , pp. 57-61.
7. Tanvi Sethi, Gagandeep," Improved Approach for Software Defect Prediction using Artificial Neural Networks ",International Conference on Reliability, Infocom Technologies and Optimization, (2016).
8. Logan Perreault, Seth Berardinelli, Clemente Izurieta, John Sheppard, Using Classifiers for Software Defect Detection, In 26th International Conference on Software Engineering and Data Engineering, (2017), pp. 2-4.
9. Hans Raukas, Some Approaches for Software Defect Prediction,Tartu (2017).
10. He, Peng, et al., An Improved Method for Cross-Project Defect Prediction by Simplifying Training Data." Mathematical Problems in Engineering 2018 (2018). Article ID 2650415, 18 pages
11. N.Kalaivani,,Dr.R.Beena. Overview of software Defect Prediction using Machine Learning Algorithms, International Journal of Pure and Applied Mathematics. (2018), Vol 118 No. 20 3863-3873 ISSN: 1314-3395.
12. Bowes, David, Tracy Hall, and Jean Petrić. "Software defect prediction: do different classifiers find the same defects?, Software Quality Journal, (2018). Vol. 26, No.2, pp. 525-552.
13. Awni Hammouri, Mustafa Hammad, Mohammad Alnabhan, Fatima Alsarayrah, Software Bug Prediction using Machine Learning Approach, International Journal of Advanced Computer Science and Applications, (2018)Vol. 9, No. 2.
14. Zhiqiang Li, Xiao-Yuan Jing , Xiaoke Zhu, Progress on approaches to software defect prediction, The Institution of Engineering and Technology. ,(2018). Vol. 12 Iss. 3, pp. 161-175.
15. Mrinal Singh Rawat, Sanjay Kumar Dubey, Software Defect Prediction Models for Quality Improvement: A Literature Study , International Journal of Computer Science Issues, ,(Sep 2012) Vol. 9 ,Issue 5,No 2.

AUTHORS PROFILE

Ms.Amutha.A, Assistant Professor, School of Computer Science and Applications, holds post graduate degree in Computer Applications and M.Phil degree in computer science from Bharathidasan University, Tamil Nadu. She has 12 years of teaching experience, teaching various subjects like Operating Systems, Unix programming, Computer Architecture and Cryptography and Network Security. She has presented and published papers in the national/international conferences and journals.

Dr. V. Arul Kumar, Assistant Professor in School of Computer Science & Applications holds doctoral degree in Computer Science from Bharathidasan University-Tamil Nadu. He has completed B. Sc (Applied Sciences – Computer Technology), M.Sc (Applied Sciences Information Technology) from K.S.R College of Technology and M.Phil in Computer Science from Bharadhidasan University, Tamil Nadu. He has 5 Years of experience in teaching and 7 years of experience in research. He has qualified in State Eligibility Test (SET) conducted by Mother Teresa Women's University. He has published 18 research articles in the various International / National Journals and conferences. His Research area includes data Mining, cloud security and cryptography.

Francis Densil Raj V is presently working as Assistant Professor in REVA University, Bangalore, India. His thirst area of Research spreads towards Networks and IoT. He is having more than 5 publications in indexed journals. Now he is working with Routing Algorithms in IoT.