A Systematic Review Based On Machine Learning Techniques for Software Defect Predication

[1]Karishma Manchanda, [2]Heena Wadhwa, Harsimran Kaur
[1]Research Scholar, Chandigarh Engineering College, Landran
[2]Department of Computer Science and Engineering, Chandigarh Engineering College, Landran
Punjab, India

Abstract-Software Defect Predication is an important step in software engineering. A meta-analysis of all relevant data, high quality primary studies of defect prediction has been used to determine that what factors are influencing the predictive performance. This paper reflects various methods of default prediction in Software module by using feature extraction and machine learning techniques and provides a systematic approach to build a defect-free system.

***Index Terms*- Feature Extraction, Machine Learning, Software Defect Predication, Software Engineering.**

I. Introduction

Defect in a software module occurs due to source code error that further produces wrong output and leads to poor quality software products. Defective software modules are also responsible for high development and maintenance cost and customer dissatisfaction. Various machine learning techniques has been used for improvement in defect predication in software modules. Data mining and machine learning are used together to study data and find previously-hidden trends or patterns within. Data mining is a computational process of discovering patterns in large datasets involving methods at the intersection of artificial intelligence, machine learning and database systems. Machine learning includes the study and construction of algorithms and takes decisions based on the qualities of the studied data using statistics and adding more advanced algorithms to achieve its goals. The defect predication in software consists of various steps which involves Data cleaning, Data extraction and Performance predications. The first step involves initial pre-processing of data, removal of constant and repeated attributes and enforcement of integrity and domain level constraints. In further step, data is extracted by feature extraction techniques which selects minimal set of features and used for classification. Machine learning algorithms are applied to build a classifier model/system. The performance of the system is then measured by various parameters like precision, recall and accuracy and compared them with the existing systems.

II. Related Work

W Afzal, Richard Torkar [8] performed an empirical comparison among various feature subset selection techniques (FSS): information gain (IG), Relief (RLF), principal component analysis (PCA), correlation-based feature selection (CFS), consistency-based subset evaluation (CNS), wrapper subset evaluation (WRP), and genetic programming (GP), on five fault prediction datasets from the PROMISE data repository. The area under the receiver operating characteristic curve (AUC) value averaged over 10-fold cross validation runs was calculated for each method on the dataset combination before and after FSS. Two machine learning algorithms, C4.5 and Naive Bayes (NB) were applied on attribute sets given by each FSS method. The results indicated that statistically there was no significant difference between the AUC values for the different FSS methods for both C4.5 and NB but a smaller set of FSS methods (IG, RLF, GP) consistently select fewer attributes and maintains the classification accuracy.

Wangshu Liu, Shulong Liu, Qing Gu, Xiang Chen, Daoxu Chen [7] proposed a robust method FECS (FEature Clustering with Selection strategies) with a certain noise tolerance capability for software fault predication. The method consists of two phases: a feature clustering phase and a feature selection phase with three different heuristic search strategies to select the most appropriate feature from each cluster. They performed a set of data pre-processing steps to guarantee the noisy free datasets and then injected class level and feature level noises simultaneously to imitate noisy datasets. Then the proposed method FECS was compared with other classical methods such as IG, CFS, and noisy datasets respectively. The result indicated the competitiveness of their approach.

Danijel Radjenovic, Marjan Hericko, Richard Torkar, Ales Zivkovic [1] aimed to identify software metrics used in software fault predication models to improve software quality by locating the faults. Many software metrics has been used in fault predication but selecting an appropriate set of metrics was important. A systematic literature review had been performed which set of metrics were suitable for fault predication. The result depicted that Object-oriented (Chidamber and Kemerer) and traditional code metrics (McCabe and Halstead) or process metrics were more successful in finding faults as compared to other traditional size and complexity metrics.

Kalai Magal. R, Shomona Gracia Jacob [3] proposed a new "Improved Random forest" algorithm to enhance the classification accuracy for software defect predication. The algorithm worked by incorporating with best feature selection algorithm and the Random Forest to give better accuracy. Correlation based Feature Subset Selection (CFS) algorithm selects the optimal subset of features. The optimal features were then fed as a part of Random Forest classification to give better accuracy in software defect

prediction. The features are selected by the CFS and utilized by Random Forest to improve the accuracy of existing Random Forest. The experiments were carried on public NASA datasets of PROMISE repository.

Mikyeong Park and Euyseok Hong [4] investigated that in supervised learning, software fault predication cannot be performed when the training data was not present. They proposed a new model based on clustering algorithm but it was difficult to decide the number of clusters. In order to solve this problem, they build an unsupervised model using clustering algorithm like EM and X-means which automatically determine the number of clusters and compared them with earlier studies. The result indicated that X-means performed far better than EM.

Venkata U.B. Challagulla, Farokh B. Bastani, I-Ling Yen [6] investigated a real time assessment technique used to classify the real time systems as faulty/fault free. Various fault free detection techniques had been used which include various machine learning methods, Statistical methods and mixed algorithms. The result indicate that there was no best technique among all the datasets but the combination of 1R classification and Instance-based Learning along with the Consistency based Subset Evaluation technique gave better results in accuracy predication.

Tracy Hall, Sarah Beecham, David Bowes [5] followed a Systematic Literature approach identified by Kitchenham and Charters and investigated that how the context of models, the independent variables used and the modelling techniques had affected the performance of software fault predication model. The result indicated that Naive Bayes algorithm based on simple modelling technique performed well.

III. Data Mining

Data mining is a computational method of discovering hidden patterns in large datasets. The overall goal of data mining is to extract information from a dataset and transform it into understandable structure for use. Data mining and machine learning are used together to study data and find previously-hidden trends or patterns within. The various phases in data mining process:

- Data selection: In this step, data relevant to the analysis tasks are retrieved from the database.
- Pre-processing: It is an essential step to analyze the data sets before mining. The target data is then cleaned removing the observations containing noisy data.
- Transformation: In this step, data is transformed into forms appropriate for mining by performing summary and aggregation operations.
- Data Mining: In this step, intelligent methods are applied in order to extract data patterns.
- Interpretation and evaluation: In this step data patterns are evaluated and knowledge is represented.

IV. Defect Predication in Software Module

When the result of the software application or module does not meet with the end user expectations, then it results into a defect. These defects occur because of an error in logic or in coding which leads to unpredicted or unanticipated results. Defective software module requires high maintenance cost and leads to poor quality. It involves various steps:

a) Data cleansing process: This is the first step in software defect predication. It involves following activities:

- Initial Pre-processing of the data
- Removal of Constant and Repeated attributes
- Replacement of Missing Values
- Enforce Integrity with Domain level constraints

b) Data Extraction process: In this step, data is extracted by using feature extraction algorithms which constructs combinations of various variables and used for dimensions reduction.

c) Performance Prediction: The model performance is measured by various parameters like accuracy, precision, recall, sensitivity etc.

V. Machine Learning

Machine learning takes decisions based on the qualities of the studied data using statistics and adding more advanced artificial intelligence heuristics and algorithms to achieve its goals. Machine learning tasks are classified into three broad categories:

- Supervised learning: The computer is presented with examples of various inputs and their desired outputs, given by the instructor and the goal is to learn a general rule to map inputs to outputs.
- Unsupervised Learning: It is basically discovering hidden patterns in data. No labels are provided to the learning algorithm, we have to find structure by own efforts from its input.
- Reinforcement Learning: A computer program interacts with a dynamic environment in which it must perform a certain goal without any instructor explicitly telling it whether it has come close to a goal.

Till now various machine learning techniques like Decision tree, Random forest, Support Vector Machine [2] are implemented for data mining. In future we are planning to construct a hybrid approach of following learning algorithms to build a software defect predication model/system.

Support Vector Machine:

Support Vector Machine (SVM) is used as a regression method for maintaining all the main features that characterize the algorithm called as maximal margin [2]. It constructs a set of hyper-planes in an infinite-dimensional space, which can be used for classification, regression, or other tasks. In other words, given labelled training data, the algorithm outputs an optimal hyper-plane which categorizes new examples. Then, the operation of the SVM algorithm is based on finding the hyper-plane that gives the largest minimum distance to the training examples.

Radial Basis Function Network:

A radial basis function network (RBF) is an artificial neural network that uses radial basis functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters. It includes the functions like approximation, time series, predication, classification, and system control. Radial basis function networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. Hidden layers provide a set of functions that constitute an arbitrary basis for input patterns and these functions are called radial basis functions.

Adaptive Boost Network:

It is a machine learning algorithm can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms is combined into a weighted sum that represents the final output of the boosted classifier.

VI. Conclusion

In this paper, we have reviewed the various machine learning techniques used for software defect predication. But still there are problems like boundary conditions, dimensionality reduction, Component learning which are not focussed yet. In future we are planning to implement a Hybrid Adaptive Boost with SVM- RBF Kernel method as a classifier model for component learning and to improve the overall performance of system.

VII. References

[1] Danijel Radjenovic, Marjan Hericko, Richard Torkar, Ales Zivkovic, "Software fault prediction metrics- A systematic literature review", Information and Software Technology, 55(8)s,pp.1397-1418,2013.

[2] J.Pradeep Kandhasamy, S. Balamurali, "Performance Analysis of Classifier Models to Predict Diabetes Mellitus", ProcediaComputerScience,Vol.No.47,pp.45–51,2015.

[3] Kalai Magal. R, Shomona Gracia Jacob, "Improved Random Forest Algorithm for Software Defect Prediction through Data Mining Techniques", International Journal of Computer, pp.0975–8887, Vol.No.23, May2015.

[4] Mikyeong Park and Euyseok Hong, "Software Fault Prediction Model using Clustering Algorithms Determining the Number of Clusters Automatically", International Journal of Software Engineering and Its Applications, Vol.No.8,pp.199-204,2014.

[5] Tracy Hall, Sarah Beecham, David Bowes, David Gray and Steve Counsell, "A Systematic Literature Review on Fault Prediction Performance in Software Engineering", IEEE Transactions On Software Engineering, Vol.N.o.38, pp.1276-1304,2011.

[6] Venkata U.B. Challagulla, Farokh B. Bastani, I-Ling Yen, "Empirical Assessment of Machine Learning based Software Defect Prediction Techniques", WORDS'05 Proceedings of the 10th IEEE International Workshop on Object-OrientedReal-TimeDependableSystems,pp.263-270,2005.

[7] Wangshu Liu, Shulong Liu, Qing Gu, Xiang Chen, Daoxu Chen, "FECS: a Cluster based Feature Selection Method for Software Fault Prediction with Noises", IEEE 39th Annual International Computers, Software & ApplicationsConferenceVol.No.2,pp.276-281,2015.

[8] Wasif Afzal, Richard Torkar, "Towards benchmarking feature subset selection methods for software fault prediction", Computational Intelligence and quantitative Software Engineering, Studies in Computational Intelligence, Vol.No.617, pp. 33-58, 2016.