International Journal of Database Theory and Application
Vol.8, No.3 (2015), pp.179-190
http://dx.doi.org/10.14257/ijdta.2015.8.3.15

A Study on Software Metrics based Software Defect Prediction using Data Mining and Machine Learning Techniques

Manjula.C.M. Prasad[1], Lilly Florence[2] and Arti Arya[3]

[1] *Associate Professor, MCA Department, PESIT, BSC, Karnataka.*
[2]*Professor, MCA Department, Adiyamman College of Engineering, Tamil Nadu.*
[3]*Professor, MCA Department, PESIT, BSC, Karnataka.*
manjulaprasad@pes.edu, lilly_swamy@yahoo.co.in, artiarya@pes.edu

Abstract

Software quality is a field of study and practice that describes the desirable attributes of software products. The performance must be perfect without any defects.Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project.The software defectprediction model helps in early detection of defects and contributes to their efficient removal and producing a quality software system based on several metrics. The main objective of paper is to help developers identify defects based on existing software metrics using data mining techniques and thereby improve the software quality.In this paper, variousclassification techniquesare revisitedwhich are employed for software defect prediction using software metrics in the literature.

Keywords:*Software Defect Prediction, Software Metrics, Classification.*

1. Introduction

In context of software engineering, software quality refers to software functional quality and software structural quality. Software functional quality reflects functional requirements whereas structural quality highlights non-functional requirements. Software metrics focus on the quality aspect of the product, process and project. In this paper the main emphasis is on software product. The objective of software product quality engineering is to achieve the required quality of the product through the definition of quality requirements and their implementation, measurement of appropriate quality attributes and evaluation of the resulting quality.

Software quality measurement [15] is about quantifying to what extent a system or software possesses desirable characteristics namely Reliability, Efficiency, Security, Maintainability and (adequate) Size. This can be performed through qualitative or quantitative means or a mix of both. In both cases, for each desirable characteristic, there are a set of measurable attributes like Application Architecture Standards, Coding Practices, Complexity, Documentation, Portability and Technical & Functional volumes. The existence of these attributes in a piece of software or system tends to be correlated and associated with this characteristic.

2. Software Metric

Software metric is a measure of a property of a piece of software or its specifications. Software metric is a way of measuring the quality of software. Software quality metrics [48] are a subset of software metrics that focus on the quality aspects of the product, process, and project.Product metrics describe the characteristics of the product such as size, complexity, design features, performance, and quality level. Process metrics can be used to improve software development and maintenance such as the effectiveness of

ISSN: 2005-4270 IJDTA

defect removal during development, the pattern of testing defect arrival, and the response time of the fix process. Project metrics describe the project characteristics and execution which includes the number of software developers, the staffing pattern over the life cycle of the software, cost, schedule, and productivity

Different product metrics are [16]

1. Chidamber and Kemerer (Chidamber andKemerer, 1994).
2. Cohesion in Methods (LCOM3) suggested by Henderson-Sellers (Henderson-Sellers, 1996).
3. The QMOOD metrics suite suggested by Bansiya and Davis (Bansiya andDavis, 2002).
4. The quality oriented extension to Chidamber&Kemerer metrics suitesuggested by Tang *et al*. (Tang *et al*, 1999).
5. Coupling metrics suggested by Martin (Martin, 1994).
6. Class level metrics built on the basis of McCabe's complexity metric (Mc-Cabe, 1976).
7. Lines of Code (LOC).

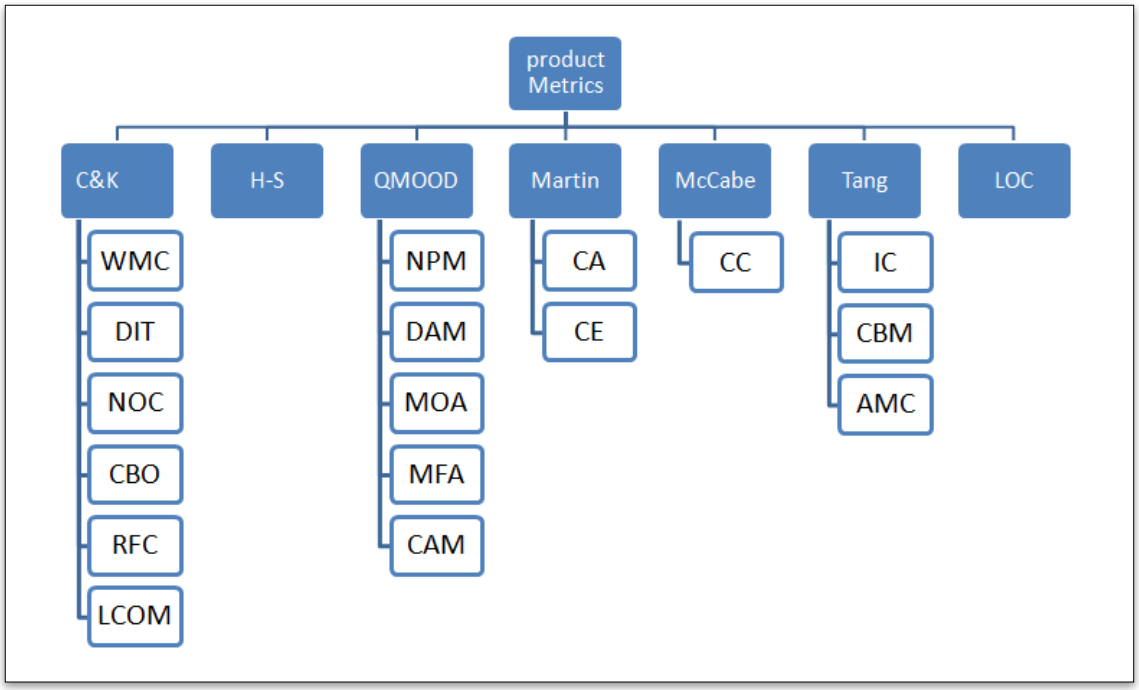

Figure1.Different Characteristics of Different Metrics [42]

WMC- Weighted Methods per Class
DIT- Depth of Inheritance Tree
NOC- Number of Children
CBO – Coupling between Object Classes
RFC- Response for a Class
LCOM-Lack of Cohesion in Methods
LCOM3- Lack of Cohesion in Methods
Ca- Afferent Couplings
Ce- Efferent Couplings
NPM- Number of Public Methods
DAM- Data Access Metric
MOA- Measure of Aggregation
MFA- Measure of Functional Abstraction

CAM- Cohesion Among Methods of Class
CC-Cyclomatic Complexity
LOC- Lines of Code
IC- Inheritance Coupling
CBM- Coupling Between Methods
AMC- Average Method Complexity
The characteristics of software metrics (features or attributes) influence the performance and effectiveness of the defect prediction model.

3. Software Defect Prediction

A software defect is an error, flaw, failure, or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. Most defects arise from mistakes and errors made by people in either a program's source code or its design, or in frameworks and operating systems used by such programs, and a few are caused by compilers producing incorrect code.

Software Defect Prediction Model refers to those models that try to predict potential software defects from test data. There exists a correlation between the software metrics and the fault proneness of the software. A Software defect prediction models consists of independent variables (Software metrics) collected and measured during software development life cycle and dependent variable (faulty or non faulty). There are different data mining techniques for defect prediction.

Data mining is the analysis step of the "Knowledge Discovery in Databases" process, or KDD, a process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further analysis.

Data Mining can be divided into two tasks: Predictive tasks and descriptive tasks. Predictive task is to predict the value of a specific attribute (target/dependent variable)based on the value of other attributes (explanatory). Descriptive task is to derive patterns (correlation, trends, and trajectories) that summarize the underlying relationship between data.

There are various data mining techniques used for software defect predictions which are discussed below.

1. Regression: It is a statistical process to evaluate the relationship among variables. It analyses the relationship between the dependent or response variable and independent or predictor variables. The relationship is expressed in the form of an equation that predicts the response variable as a linear function of predictor variable. [42, 24, 51, 25]
Linear Regression: $Y=a+bX+u$

2. Association Rule Mining: It is a method for discovering interesting relationships between variables in large databases. It is about finding association or correlations among sets of items or objects in database. It basically deals with finding rules that will predict the occurrence of item based on the occurrence of other items. [11, 17, 40,26]

3. Clustering: Clustering is a way to categorize a collection of items into groups or clusters whose members are similar in some way. It is task of grouping a set of items in such a way that items in the same cluster are similar to each other and dissimilar to those in other clusters. [27, 34, 17, 30]

4. Classification: It consists of predicting a certain outcome based on a given input. Classification technique use input data, also called training set where all objects are already tagged with known class labels. The objective of classification algorithm is to analyze and learns from the training data set and develop a model. This model is then used to classify test data for which the class labels are not known. [43, 27, 30,6, 22]. The various classification techniques are given below.

a. Neural Networks: Neural Networks are the non linear predictive models which can learn through training and resemble biological neural networks in structure. A neural network consists of interconnected processing elements called neurons that work together in parallel within a network to produce output. [22, 21, 47, 42]

b. Decision Trees: A decision tree is a predictive model which can be used to represent both classification and regression models in the form a tree structure. It refers to a hierarchical model of decisions and their consequences. It is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf nodes represent a classification or decision. [39, 31, 37]

c. Naive Bayes: It is based on Bayes theorem with independence assumption between predictors. Naive Bayes Classifier is based on the assumption that the presence or absence of a particular feature of a class in not related to the presence or absence of any other feature. [21, 14, 28]

d. Support Vector Machines: SVM are based on the concept of decision planes that define decision boundaries. A decision plane is the one that separates between a set of objects having different class membership. SVM is primarily a classifier method that performs classification task by constructing hyper plane in a multidimensional space that separates cases of different class labels. It supports both regression and classification. [50, 10, 29]

e. Case Based Reasoning: Case based reasoning means solving new problems based on the similar past problems and using old cases to explain new situations. It works by comparing new unclassified records with known examples and patterns. A simple example of a case based learning algorithm is k-nearest neighbour algorithm. It is simple algorithm that stores all available cases and classifies new cases based on a similarity measure i.e. distance function. [39]

Table1 shows the comparative analysis of Algorithms for Supervised Classification Type.

Table 1: Comparative Analysis of Supervised Classification Type [13]

Algorithm	Pros	Cons
BR	Fits Calculation diagonal matrices	No tag correlations performed explicitly
Ada boost	Excellent for sorting better accuracy	Generalizing results in decreased performance
Back Propagation	Learning iteratively. More capacity of generalization	Computationally complex presented by the algorithm
C4.5	Based on decision trees, improving accuracy and prediction. Easy to understand, popular and powerful	Not takes correlation between classes

Table2shows the comparative analysis of Algorithms for Semi-Supervised Classification Type.

Table 2. Comparative Analysis of Semi-Supervised Classification Type[13]

Algorithm	Pro's	Con's
Multi-label classification by constrained non-negative matrix factorization	Adaptable to semi-supervised environments along with the representation of documents in rank matrix factorization	There is a strong influence from two parameters on the performance: latent variables and tuning

	using the representation of documents in rank matrix factorization using the non-negative.	parameters.
Graph-based SSL with multi-label	Effective use of large amounts of unlabelled data and the ability to exploit the relationships between labels	Most of the time is used for video files. It does not adapt well to texts.
Multi-label learning by using dependency among labels	Improving accuracy by configuring SSL	Time increment for large data sets
Semi-supervised multi-label learning by solving a Sylvester	Use of large amounts of unlabeled data as well as the ability to exploit the relationship between labels. Significant improvement in the precision.	May become slow when using large data sets
Semi-supervised non-negative matrix factorization	Using NMF in conjunction with SSL allows the extraction of the most discriminating than if MFN were used.	Computational Complexity

4. Software Defect Prediction (SDP) using Different Classification Techniques

A survey is conducted to help developers identify defects based on existing software metrics using data mining techniques especially Classification and there by improve software quality which leads to reduction in the software development cost in the development and maintenance phase.Different classification techniques have been surveyed with different data sets.

4.1SDP using Supervised Learning

The various Supervised Learning techniques are discussed in this section.

4.1.1 SDP using Bayesian Network

Yuan Chen, *et.al*[38] have surveyed the different data mining classification techniques for software defect prediction. They proposed a new model based on Bayesian network and PRM to predict the software defect and manage.

Hassan Najadat and IzzatAlsmadi[33]Proposed a new model based on Ridor algorithm to predict fault in modules. They also tested the different classification techniques on the data sets provided by NASA. The results shown that Ridor algorithm is better than the existing technique in terms of accuracy and extraction of number of rules.

Ahmet Okutan,OlcayTanerYıldız [20],Introduced a new two metrics NOD, for the number of developers and LOCQ for source code quality apart from the metrics which is available in Promise data repository. Using Bayesian network classifier experimental shows that NOC &DIT have very limited and untrustworthy. LOCQ is more effective like CBO & WMC. NOD metric showed that there is a positive correlation between the no of developers and extent of defect prunes. LOC is proved to be one of the best metric for quick defect prediction. LCOM3 & LCOM have less effective compared to LOC,CBO,RFC, and LOCQ&WMC.

Thair Nu Phyu [39] reviewed on various classification techniques such as decision tree induction, Bayesian networks, k-nearest neighbour classifier, case-based reasoning,

genetic algorithm and fuzzy logic techniques. The results found that there is no proper info that which is the best classifier. Several of the classification methods produce a set of interacting loci that best predict the phenotype. However, a straightforward application of classification methods to large numbers of markers has a potential risk picking up randomly associated markers.

Wen Zhang et.al [4] proposed Bayesian Regression Expectation Maximize algorithm for software effort prediction and two embedded strategies handle missing data. They used the method of ignoring the missing data in an iterative manner in the predictive model. Here they have used data sets such as ISBSG and CSBSG. When there are no missing data BREM outperforms CR, BR, and SVR& M5. When there are missing data BREM with MDT and MDI outperforms imputation technique includes MI,BMI,CMI,and Mini& M5. BRM is used for software prediction and MDI used for finding missing values embedded with BREM.

Arvinder Kaur and InderpreetKaur[7] , they have tried to find the quality of the software product based on identifying the defects in the classes. They have done this by using six different classifiers such as Naive base, Logistic regression, Instance based (Nearest- Neighbour), Bagging, J48, Decision Tree, Random Forest. This model is applied on five different open source software to find the defects of 5885classes based on object oriented metrics. Out of which they found only Bagging and J48 to be the best.

K.Sankar et.al [10], proposed a system which overcomes the problem of insufficiency in accuracy and use of large number of features. This paper proposed Feature selection techniques to predict faults in software code and it also measure the software code and performance of Naive based and SVM classifier. The accuracy is measured by F-mean metric.

4.1.2. SDP using Ensemble Method/ Random Forests

Issam H*et.al*[3] have proposed a two-variant ensemble learning classifier which shows that greedy forward selection is better than correlation forward selection. Further they proposed a model APE with seven different classifiers which results much better when compared to weighted SVM's and random forests. Further thy enhanced the version of APE with greedy forward selection to produce higher AUC measures for the different data sets. The results shown stronger robustness to redundant and irrelevant features.

Renqing Li &Shihai Wang[12] predicted defects on imbalanced data sets. C4.5, SVM, KNN, Logistic, Naïve Base, Ada boost &smooth boost models were tested on imbalanced data sets of NASA's MDP. The results found that Smooth boost found to be the best defect predictor when compared to the others.

Yan ma et.al [47] proposed a model based on random forests. This is applied on five case studies based on NASA data sets. The results found was better than the result obtained by logistic regression, discriminate analysis and the algorithms in two machine learning software packages . Instead of generating one decision tree, this methodology generates hundreds or even thousands of trees using subsets of the training data. Hence classification accuracy of random forests is more significant over other methods in larger data sets.

C.Chung and S.Dhall [29] proposed a various classification methods to predict software defect. Here Three types of classifier such as J48, Random Forest and Naive Bayesian Classifier is applied on various real time data sets of NASA to evaluate the data sets based on different criteria like ROC, Precision, MAE, RAE *etc*.

4.1.3. SDP using Support Vector Machine

Sonali Agarwal and DivyaTomar[9] have proposed a feature selection based Linear Twin Support Vector Machine (LSTSVM) model to predict defect prone software modules. F-score technique is used for software defect prediction based on various software metrics. This model is applied on PROMISE data sets and compared with the other existing models. The results say that the performance of the new model is better than the existing machine learning models.

CagatayCatal [23] proposed four semi-supervised classification methods such as Low-density separation (LDS), support vector machine (SVM), expectation-maximization (EM-SEMI), and class mass normalization (CMN) for semi-supervised defect prediction. They applied 4 types of ssc on NASA datasets. The results showed that SVM & LDS are better than CMN and EM-SEMI. LDS performs much better than SVM for a large data set.

Karim O. Elish, Mahmoud OElish[45] proposed SVM is the model and compared with the eight different statistical and Machine learning models The compared models are two statistical classifiers techniques: (I)Logistic Regression (LR) and (ii) K-Nearest Neighbour (KNN); two neural networks techniques: (I) Multi-layer Perceptrons (MLP) and (ii) Radial Basis Function(RBF); two Bayesian techniques: (I) Bayesian Belief Networks (BBN) and (ii) Naı¨ve Bayes (NB); and two tree structured classifiers techniques: (I) Random Forests (RF) and (ii) Decision Trees (DT) using four NASA data sets. The results found that SVM is the better model when compared to the other models.

David Gray*et.al* [41] proposed a work using the static code metrics for a collection of modules contained within eleven NASA data sets are used with a Support Vector Machine classifier. A rigorous sequence of pre-processing steps were applied to the data prior to classification, including the balancing of both classes (defective or otherwise) and the removal of a large number of repeating instances. The Support Vector Machine in this experiment yields an average accuracy of 70% on previously unseen data.

4.1.4. SDP using Decision Tree

GolnoushAbaei·AliSelamat [19], In this paper many different machine learning techniques such as decision trees, decision tables, random forest, neural network, Naïve Bayes and distinctive classifiers of artificial immune systems (AISs) such as artificial immune recognition system, CLONALG and Immunos. Experiment is performed on four public NASA datasets which are different in size and number of defective data. The results obtained are ran-dom forest provides the best prediction performance for large data sets and Naïve Bayes is a trustable algorithm for small data sets even when one of the feature selection techniques is applied. Immunos99 performs well among AIS classifiers when feature selection technique is applied, and AIRS Parallel perform better without any feature selection techniques.

Thair Nu Phyu [39] reviewed on various classification techniques such as decision tree induction, Bayesian networks, k-nearest neighbour classifier, case-based reasoning, genetic algorithm and fuzzy logic techniques. The results found that there is no proper info that which is the best classifier. Several of the classification methods produce a set of interacting loci that best predict the phenotype. However, a straightforward application of classification methods to large numbers of markers has a potential risk picking up randomly associated markers.

Ching-Pao Chang*et al.*[40] proposed approach, Action-Based Defect Prediction (ABDP),which uses the classification with decision tree technique to build a prediction model, and performs association rule mining on the records of actions and defects. The association rule mining finds the maximum rule set with specific minimum support and confidence and thus the discovered knowledge can be utilized to interpret the prediction models and software pro-cess behaviours. It is used todiscover defect patterns, and multi-

interval discretization to handle the continuous attributes of actions.

4.2 SDP using Semi-supervised Learning

Ming Li, *et al.*proposed [32] a sample based methods for software defect prediction. Three methods such as random sampling with conventional machine learners, random sampling with a semi-supervised learner and active sampling with active semi-supervised learner. They applied a semi-supervised learning method called ACoForest to build a classification model based on a sample and the remaining un-sampled mod-ulesthey also proposed a novel active semi supervised method called AcoForest which can select unsampled modules and experimented on Promise data sets and found to be the best method. Experimental results show that size does not affect the defect prediction.

CagatayCatal [23] proposed four semi-supervised classification methods such as Low-density separation (LDS), support vector machine (SVM), expectation-maximization (EM-SEMI), and class mass normalization (CMN) for semi-supervised defect prediction. They applied 4 types of ssc on NASA datasets. The results showed that SVM & LDS are better than CMN and EM-SEMI. LDS performs much better than SVM for a large data set.

GolnoushAbaei,*et al* [1] proposed a semi- supervised HYSOM model (hybrid self-organizing map) in order to detect defects with a high accuracy and improve detection model generalization ability HYSOM model will predict the label of modules in a semi-supervised manner. This is applied on eight industrial data sets from NASA and Turkish data set. It can also be used as an automated tool to predict defects in less time for project managers , software developers and Testers.

4.3. SDP using Unsupervised Learning

C.Chung and S.Dhall[29] proposed a various classification and clustering methods to predict software defect. The various data mining classifier algorithms namely J48, Random Forest, and Naive Bayesian Classifier (NBC) are evaluated based on various criteria like ROC, Precision, MAE, RAE *etc*. Clustering technique is later applied on different data set of NASA using k-means, Hierarchical Clustering and Make Density Based Clustering algorithm. Results are evaluated based on criteria like Time Taken, Cluster Instance, Number of Iterations, Incorrectly Clustered Instance and Log Likelihood *etc*.

Dhiman,*et al.* [53] proposed a model where in it will categorize the software defects using some clustering approach and then the software defects are measured in each clustered separately. This system will analyze the software defect and its integration with software module.

4.4. SDP using Machine Learning Algorithm

Xiao-YuanFing,*et.al* [8] have tried to model the effective , efficient and low computational burden using advanced machine learning technique such as collaborative representative classification. The new model proposed by them is CSDP which is used to predict defect in a very efficient manner.

KehanGao&Taghi M [35] experimented on promise repository based on criteria 1) Feature selection based on sampled data, and modelling based on original data, 2) feature selection based on sampled data and modelling based on sampled data and 3) feature selection based on sampled data, and modelling based sample data. The experimental results showed that the 1st criteria is the best compared to the others in defect prediction.

S.Bibi*et al* [44] proposed a RVC model for finding the defects in the software by using symbolic learning algorithms. They have compared the model with several machine learning algorithms in two software data sets and the results found were better regression error than the standard regression approaches on both data sets. Apart from finding the

faults it also produces an associated interval of values within which this estimate lies with a certain confidence.

5. Conclusion and Future Scope

Software quality is the degree of conformance to explicit or implicit requirements and expectations. A software metric is a quantitative measure of a degree to which a software system or process possesses property with no defects. Hence, Software defect prediction model helps in early detection of defects using Classification Technique. In this paper we have discussed the various classification techniques such as Supervised, Un-supervised and Semi-supervised, which are applied on various datasets based on existing software metrics. In future we will be comparing the results of Supervisedclassification techniques on different datasets and open source projects to analyze the best classification technique to predict the defect in order to evolve a good software quality product.

References

[1] G.Abaei[a], A.Selamat[a], H.Fujita[b], "An empirical study based on semi-supervised hybrid self-organizing map for software fault prediction", Knowledge-Based Systems, vol. 74, **(2015)**, pp. 28-39.
[2] R. Malhotra, "A systematic review of machine learning techniques for software fault prediction",Applied Soft Computing, vol. 27, **(2015)**, pp. 504-518.
[3] I. H. Laradji, M.Alshayeb, L.Ghouti, "Software defect prediction using ensemble learning on selected features. Information and Science Technology", vol. 58, **(2015)**, pp. 388-402.
[4] W. Zhang, Y. Yang, Q. Wang, "Using Bayesian Regression and EM algorithm with missing handling for software effort prediction", Information and software technology, vol. 58, **(2015)**, pp. 58-70.
[5] P. He, B. Li, X. Liu, J. Chen, Y. Ma, "An empirical study on software defect prediction with a simplified metric set", vol 59, **(2015)**, pp. 170-190.
[6] V. Ajay Prakash, D. V. Ashoka, V. N. ManjunathAradya, "Application of Data Mining Techniques for Defect Detection and Classification", Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014, Advances in Intelligent Systems and Computing, vol. 327, **(2015)**, pp. 387-395
[7] A. Kaur and I. Kaur, "Empirical Evaluation of Machine Learning Algorithms for Fault Prediction", LectureNotes on Software Engineering, vol. 2, no. 2, **(2014)**.
[8] X. Yuan, H.W. Zhang, S. Ying,F. Wang, "Software defect prediction based on collaborative representation classification", Proceedings in ICSE Companion 2014, 36th International Conference on Software Engineering, pp. 632-633.
[9] S. Agarwal and D.Tomar, "A Feature Selection Based Model for Software Defect Prediction", International Journal of Advanced Science and Technology, vol.65,**(2014)**, pp. 39-58.
[10] K. Sankar, S. Kannan and P.Jennifer, "Prediction of Code Fault Using Naive Bayes and SVM Classifiers Middle-East Journal of Scientific Research", vol. 20, no. 1, **(2014)**, pp.108-113.
[11] G.Czibula, Z. Marian, I. G.Czibula, "Software defect prediction using relational association rule mining, Information Sciences", vol. 264, no. 20 **(2014)**, pp. 260-278.
[12] R. Li, S.Wang, "Ann Empirical Study for Software Fault-Proneness Prediction with Ensemble Learning Models on Imbalanced Data Sets", Journal of Software, vol. 9, no.3,pp. 697-704,**(2014)**.
[13]M. Barcelo-Valenzuela, M. Romero-Ochaoa, A. Perez-Soltero, G. Sanchez-Schmitz, "Knowledge Sources and Automatic Classification: A Literature Review", International Journal of Business, Humanities and Technology, vol. 4, no. 1, **(2014)**.
[14] L. Li, H. Leung, "Bayesian Prediction of Fault-Proneness of Agile-Developed Object-Oriented System:Lecture Notes", Business Information Processing, vol. 190, **(2014)**, pp. 209-225.
[15] The Global Conference for Wikimedia,**(2014)**; London.
[16] L. Madeyski, M.Jureczko, "Which process metrics can significantly improve defect prediction models?", An empirical study,**(2014)**.
[17] D.Mehta, "A Comparative study of Techniques in Data Mining", by Manika Verma[1], International Journal of Emerging Technology and Advanced Engineering, vol. 4, no. 4, **(2014)**.
[18] P. Reena, R. Binu, "Software Defect Prediction System –Decision Tree Algorithm With Two Level Data Pre-processing", International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 3, **(2014)**.
[19] G.Abaei, A.Selamat, "A survey on software fault detection based on different prediction approaches", Vietnam Journal of Computer Science, **(2014)**, vol. 1, no. 2, pp. 79-95.
[20] A.Okutan, O. T.Yildiz, "Software defect prediction using Bayesian networks", Empirical Software Engineering, **(2014)**, vol. 19, no. 1, pp. 154-181.

[21] A.TosunMisirli, A. se Ba¸ S.Bener,"A Mapping Study on Bayesian Networks for Software Quality Prediction", Proceedings of the 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, **(2014)**.
[22] R.Kalsoom, M. Qureshi, "Application and Verification of Algorithm Learning Based Neural Network",arXiv preprint arXiv:1406.2614, **(2014)**, arxiv.org.
[23] C. Catal, "A Comparison of Semi-Supervised Classification Approaches for Software Defect Prediction", Journal of Intelligent Systems, vol. 23, no. 1, pp. 75-82,**(2013)**.
[24] R.Goyala, P.Chandraa, Y. Singha, "Suitability of KNN Regression in the Development of Interaction Based Software Fault Prediction Models", IERI Procedia, International Conference on Future Software Engineering and Multimedia Engineering, Elsiever, vol 6, pp. 15-21, **(2013)**,.
[25] G.Scanniello, C.Gravino, A.Marcus,T.Menzies,"Class level fault prediction using software clustering, Automated Software Engineering (ASE)", 2013 IEEE/ACM 28th International Conference, **(2013)**.
[26] B. V. Balaji1, V.Venkateswara Rao2, "Improved Classification Based Association Rule Mining", International Journal of Advanced Research in Computer and communication Engineering, vol. 2, no. 5, **(2013)**.
[27] R. M. Rahman, F. Afroz,"Comparison of Various Classification Techniques Using Different Data Mining Tools for Diabetes Diagnosis",Journal of Software Engineering and Applications, **(2013)**, vol.6, pp.85-97
[28] T. Angel Thankachan[1], K. Raimond[2], "A Survey on Classification and Rule Extraction Techniques for Data mining",IOSR Journal of Computer Engineering ,vol. 8, no. 5,**(2013)**, pp. 75-78.
[29] A. Chug[1] and S. Dhall[1], "Software Defect Prediction Using Supervised Learning Algorithm and Unsupervised Learning Algorithm",The Next Generation Information Technology Summit (4th International Conference),**(2013)**,pp.1-6.
[30] "Software defect prediction using supervised learning algorithm and unsupervised learning algorithm", Confluence 2013: The Next Generation Information Technology Summit (4th International Conference), **(2013)**.
[31] M. Surendra Naidu, "Classification of Defects in Software Using Decision Tree Algorithm", International Journal of Engineering Science and Technology (IJEST), **(2013)**.
[32] M. L., H. Zhang, R. Wu, Z.-H. Zhou, "Sample-based software defect prediction with active and semi-supervised learning", Automated Software Engineering , **(2012)**, vol. 19, no. 2, pp. 201-230
[33] H.Najadat and I.Alsmadi, "Enhance Rule Based Detection for Software Fault Prone Modules", International Journal of Software Engineering and Its Applications, vol. 6, no. 1, **(2012)**.
[34] S. Kaur, and D. Kumar, "Software Fault Prediction in Object Oriented Software Systems Using Density Based Clustering Approach", International Journal of Research in Engineering and Technology (IJRET) vol. 1, no. 2,**(2012)**.
[35] K. Gao, T..M.Khoshgoftarr, "Software Defect Prediction for high- dimensional and class-imbalanced data", 23rd International Conference on Software Engineering & Knowledge Engineering (SEKE'2011), Eden Roc Renaissance, **(2011)**Miami Beach, USA.
[36] B. Ma, D. Karel, V. Jan, B. Bart, "Software defect prediction based on association rule classification", Research Center for Management Informatics (LIRIS), Leuven, **(2011)**.
[37] C. Catal, U.Sevim, B. Diri,"Practical development of an Eclipse-based software fault prediction tool using Naive Bayes algorithm", Elsevier,**(2011)**.
[38] Y. Chen, P. Du,Xi , X.-H. Shen, "Research on Software Defect Prediction Based on Data Mining", Computer and Automation Engineering(ICCAE), 2nd International Conference, **(2010)**, vol. 1, pp. 563-567.
[39] T. Nu Phyu, "Survey of Classification Techniques in DataMining", International MultiConference of Engineers and Computer Scientists, **(2009)**; Hong Kong.
[40] C.-P.Chang [a,*], C.-P.Chu [a], Y.-F.Yeh[b], "Integrating in-process software defect prediction with association mining to discover defect pattern", Information and Software Technology ,vol. 51, no. 2, **(2009)**, pp. 375-384.
[41] D.Gray,D. Bowes, N. Davey, Y. Sun, "Bruce Christianson, Using the Support Vector Machine as a Classification Method for Software Defect Prediction with Static Code Metrics",11th International Conference, EANN 2009, **(2009)**; London, UK.
[42] M. Jureczko, "Significance of Different Software Metrics in Defect Prediction", Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology,WybrzeżeWyspiańskiego vol. 27, pp.50-370.
[43] S. Lessmann,B.Baesens, C.Mues, and S. Pietsch,"Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings", IEEE Transactions on Software Engineering, **(2008)**.
[44] S. Bibi', G. Tsoumakas, I. Stamelos, I. Vlahavas, "Regression via Classification applied on software defect estimation",Elsiever,vol. 34, no. 3,**(2008)**, pp. 2091-2101.
[45] K. O. Elish, M. O. Elish, "Predicting defect-prone software modules using support vector machines" ,Elsevier, vol. 81, no. 5, **(2008)**.

[46] E. O. Costa, G. A. de Souza, A. T. R.Pozo, and S. R.Vergilio, "Exploring Genetic Programming and Boosting Techniques to Model Software Reliability", IEEE Transaction on Reliability, vol. 56, no. 3, **(2007)**.
[47] Y. Ma,C. Bojan,"Singh:Robust prediction of fault-proneness by random forests ,Software Reliability Engineering", ISSRE 2004. 15th International Symposium,**(2004)**,pp. 417-428.
[48] www.pearsonhighered.com/samplechapter4/software quality metrics overview
[49] G.Mauša, mag. ing. el., "Search Based Software Engineering and Software Defect Prediction", University of Rijeka - Faculty of Engineering, Vukovarska 58, HR-51000 Rijeka, Croatia
[50] G. H. Jozsefvalyon, "Least Squares Support Vector Machines for Data Mining", Budapest University of Technology and Economics, Department of Measurement and Information Systems, published in Neural Networks, Proceedings, IEEE International Joint Conference, **(2003)**.
[51] S.Bibi, G.Tsoumakas, I.Stamelos, I. Vlahavas, "Software Defect Prediction Using Regression via Classification", Department of Informatics, Aristotle University of Thessaloniki,54124 Thessaloniki, Greece.
[52] G.Bhavya, "Improving the Fault Prediction inOo Systems Using ANN with Firefly Algorithm",International Journal of Innovative Research in Science & Engineering, pp. 2347-3207.
[53] P.Dhiman, M.C. Manish,"A Clustered Approach to Analyze the Software Quality Using Software Defects, Advanced Computing & Communication Technologies (ACCT)", 2012 Second International Conference,**(2012)**.

Authors

Manjula.C.M. Prasad, she is working as an Associate Professor at PESIT, BSC, Bangalore,India. She is persuing her PhD from Bharathiyar University. Her research area is Software Engineering and Data Mining.

Lilly Florence, she is working as a Professor at MCA department, Adiyamman College of Engineering, Tamilnadu, India. She has completed her PhD in Computer Science from Mother Theresa University, Tamilnadu. Her area of interest include Artificial Intelligence, Software Reliability and Data Mining *etc*.

Arti Arya, she is working as a Professor and Head of the Department of MCA, at PESIT, BSC, Bangalore, India. She has Bachelor's and Master's in Mathematics from Delhi University.She completed her PhD in Computer Science Engineering from MDU in 2009. She has more than 20 publications in reputed Conferences and Journals. Her areas of interest include Data Mining, Text Mining, Artificial Intelligence, Knowledge Management, Big Data Analytics *etc*.

