Comparative Performance of Fault-Prone Prediction Classes with K-means Clustering and MLP

Satwinder Singh
Central University of Punjab, Bathinda
satwinder.singh@bbsbec.ac.in

Rozy Singla
Shankara Institute of Technology, Jaipur
rozysingla92@gmail.com

ABSTRACT
Software defect in today's era is most important in the field of software engineering. Most of the organizations used various techniques to predict defects in their products before they are delivered. Defect prediction techniques help the organizations to use their resources effectively which results in lower cost and time requirements. There are various techniques that are used for predicting defects in software before it has to be delivered. For example clustering, neural networks, support vector machine (SVM) etc. In this paper two defect prediction techniques: - K-means Clustering and Multilayer Perceptron model (MLP), are compared. Both the techniques are implemented on different platforms. K-means clustering is implemented using WEKA tool and MLP is implemented using SPSS. The results are compared to find which algorithm produces better results. In this paper Object-Oriented metrics are used for predicting defects in the software.

Keywords
Object-Oriented Metrics, Defect prediction, K-means Clustering, Neural Network, Weka.

1. INTRODUCTION
Standish Group carried a survey on cost of software projects. They concluded that the cost of software increases by 90% and the schedule of project exceeded by 222%. Data is collected from 8000 projects for this survey. According to this this survey it is visualized that it is important to measure the software early and necessary actions should be taken before the product delivers. These actions may be bug finding, defect prediction, defect correction etc. In software industry, a metrics program for software projects is usually seen unnecessary, but when things go bad then the practitioners start to stress on these metrics programs. These metrics programs help to evaluate the performance of software before the delivery [1].

If there is a well established metrics program presents than cost and schedule of software project can be effectively predicted. The measured metrics give better detection during development phase of software. One of the most popular techniques for defect prediction is testing. But the testing is too complicated and expensive when the size of project grows. Clusters can be created to group the defects according to their types. If there is any relationship between software design metrics and defects properties, then it is possible to predict similar type of faults or defects in other parts of the software [2].

There are many methods used for defect prediction such as classification, statistical procedures, machine learning methods (Artificial Neural Network (ANN), Genetic Algorithm (GA) etc.) etc. Defect prediction is most challenging part in software development life cycle. Classification is the most popular method for predicting the defects in the software. Classification is used to predict discrete or unordered labels and defect prediction is used to predict continuous valued functions. Defect prediction plays as essential role in software quality and software reliability. Defect prediction may increase the cost of software development. It may not eliminate all the defects but minimizes the number of defects and their impact on the software reliability [3].

Fault prediction is a mechanism used in software development life cycle to reduce the software failure and identify fault-prone modules. Fault prediction not only increases the quality of monitoring during software development but also gives suggestions for suitable verification and validation approaches that eventually lead to improvement of efficiency and effectiveness of fault prediction [5].

Present day software development is mostly based on Object-Oriented (OO) paradigm. The quality of OO software can be best assessed by the use of software metrics. A number of metrics have been proposed by researchers and practitioners to evaluate the quality of software [4].

 ACM 978-1-4503-3962-9/16/03…$15.00
DOI: http://dx.doi.org/10.1145/2905055.2905123.

The k-means clustering is a useful algorithm in generating good quality outcome for various realistic applications. The k-means clustering intend to categorize *n* objects into *k* clusters in which each object belongs to the cluster with the nearest centroid serving as a model of the cluster. It is a partitional clustering approach which is simple and easy to understand. Each cluster produced using k-means clustering, has a centroid i.e. center point. Every object is categorized into the cluster with the nearest centroid. The number of clusters must be specified.

There are different clustering algorithms available, such as:basic sequential algorithmic scheme, Complete-linkage clustering, DBSCAN, Expectation–maximization algorithm, Fuzzy clustering, Hierarchical clustering, Mean shift, Nearest-neighbor chain algorithm, Single-linkage clustering, Spectral clustering etc.

Performance evaluation parameters:-

The following sub-sections give the basic definitions of the performance parameters used for fault prediction.

Table 1: Confusion Matrix

	Non-Faulty	**Faulty**
Non-Faulty	True Negative (TN)	False Positive (FP)
Faulty	False Negative (FN)	True Positive (TP)

The confusion matrixes are categories into four categories:

i. True positives (TP) are the number of modules correctly classified as faulty modules.

ii. False positives (FP) refer to not-faulty classes incorrectly labeled as faulty classes.

iii. True negatives (TN) correspond to not-faulty modules correctly classified as such.

iv. Finally, false negatives (FN) refer to faulty classes incorrectly classified as not-faulty classes.

These are the performance parameter used to measures the classification techniques.

Precision:-

It is used to measure the degree to which the repeated measurements under unchanged conditions show the same results. It is also called as positive predictive value. Value near one is good. Precision is related to random errors.

$$Precision = \frac{TP}{FP + TP}$$

Specificity:-

It is also known as True Negative Rate (TNR). It defines how the faulty classes are classified as faulty by the predictor. It is defined as:

$$\text{Specificity} = \frac{TN}{FP+TN}$$

Accuracy:-

Accuracy measure is the proportion of predicted fault prone modules that are inspected out of all modules. Model is said to perfect if it has accuracy value 1. Accuracy includes both trueness and precision. It is defined as:

$$\text{Accuracy} = \frac{TN+TP}{TP+TN+FP+FN}$$

TP Rate:-

The True positive rate measures the proportion of faults correctly specified as faulty modules in the dataset and is complementary to the false negative rate. It is also called the Sensitivity or recall**.**

$$\text{True positive rate} = \frac{TP}{TP+FN}$$

FP Rate:-

The False Positive rate specifies the tendency to predict the non-faulty modules as faulty.

$$\text{False positive rate} = \frac{FP}{FP+TN}$$

AUC (Area Under Curve):-

The Area under Receiver Operating Characteristics (AUC) curve presents the association of True positive rate and False positive rate.

2. LITERATURE SURVEY

Qinbao Song *et al.* [6] focuses on the classification of software components according to their defect-proneness i.e. to classify them as fault-prone or non-defect prone. They argued on *how* the attributes are used to build predictors and *which* particular attributes are used for defect prediction. They focused on *which* i.e. the selection of attribute data set used for learning or training. They designed a framework that includes a scheme evaluation and defect prediction. In scheme evaluation various learning or training strategies are used. In defect prediction stage, the best suited learning scheme is selected to build the prediction model. They used ROC (Receiver Operating Characteristic) as performance measurement parameter to calculate AUC (Area under Curve). They have used 12 different learning schemes to predict defect-prone modules. They come up with the result that different learning schemes are selected for different data sets.

Norman E. Fenton and Martin Neil [7] presented a critical review of software prediction techniques. In this review they show that size, complexity, testing and quality metrics can be used for estimating the defects. There is a direct dependency between the size metrics and defects and defects can be described as a function of size. One class of testing metrics that generates good results for predicting defects are the so called test coverage measures. Prediction can also be done with the help of process quality metrics. The simplest process quality metrics is SEI Capability Maturity Model (CMM) ranking. There are various other techniques that can be used for prediction such as neural network, clustering, regression etc. which uses these metrics. They represented some issues affecting the software engineering community with respect to defect prediction. As the relationship between defects and failures are unknown, size and complexity metrics can't be used as a sole in the defect prediction. They proposed a prediction technique which is based on Bayesian Belief Network (BBN). A BBN is a graphical network which represents probabilistic relationship among variables such as relationship between defects and software metrics. The advantage of BBN is that it can predict events based on uncertain data and ability to represent and manipulate complex models. They give theoretical comparison which is not tested by them.

Yan Ma *et al.* [8] proposed a fault prediction strategy using a modified random forest

method. They used five NASA datasets which varies in size. These datasets contain a little number of defect samples in the training set. They used balanced Random Forest as fault prediction methodology. They compare their results with data mining software packages like WEKA, See5 and SAS. Random Forest algorithm builds a classification tree based on impurity (heterogeneity) function such as Entropy-based function or Gini-index function. They used PD (Probability of Detection), PF (Probability of False Alarms), Accuracy, TNR (True Negative Rate), G-mean and F-measure values for comparison. The results show that Balanced Random Forest is better than Random Forest for classification.

Name	Number of Classes	Number of Defect Classes	% Defects
Licq	280	126	45
Seamonkey 1.0.1	4103	47	1.145

Malkit Singh and Dalwinder Singh Salaria [9] used Software metrics such as file-level, class-level, component-level, method-level, process-level and quantitative values-level metrics etc. to find the software defects. Various methods such as statistics, machine learning, and machine learning along with statistical methods are used for calculating defects. Researchers can also apply a machine learning approach on real-time software systems for defect prediction. They used Ant 1.7 data set from PROMISE repository for the experiment. They divide data as 85% data for training and 15% data for testing. The data is trained with Levenberg-Marquardt (LM) algorithm resulting in 88% accuracy.

Jain and Dubes [10] defined the main steps of K means algorithm as: 1.Select an initial partition with K clusters; repeat steps 2 and 3 until cluster membership stabilizes.2. Generate a new partition by assigning each pattern to its closest cluster center.3.Compute new cluster centers. The K-means algorithm depends upon three user-specified parameters: number of clusters K, cluster initialization, and distance metric. Choosing value of K is most censorious. There are different extensions of K-means algorithm. The enlarged heuristics are tackled by some of these extensions that involve the minimum cluster size and merging and splitting clusters. There are two popular alternatives of K-means in pattern recognition literature that are ISODATA proposed by **Ball and Hall [11] and FORGY Forgy** [12].

3. Proposed Model and Methodology:-

Data Set:-

System chosen for defect prediction analysis is Licq (an instant messaging or communication client for the UNIX). Licq bug database was collected from the GitHub[1] community. Licq is smaller system with 280 classes. One Mozilla Seamonkey Version 1.0.1 is open source system was used for analysis in this study. Whereas, the bug database required for each version was collected from the Bugzilla[2] system. All errors (bugs) that have been found in the lifetime of the Mozilla Firefox project are stored in the Bugzilla database. Bugzilla database has detailed information that includes the release number, error severity, and summary of errors.

Table 2: Dataset Details

Methodology: - Research community shows that better data mining technology is not leading to better defect predictors. High accuracy in prediction of various machine learning models studied by various scientific communities motivates the current study to pursue the problem with it. A learner framework for defect prediction is trying to be proposed with the machine learning model in the current study. Machine learning focuses on prediction, based on known properties learned from the training data. There are various machine learning models as Decision Tree based methods, Linear regression based methods, Neural network, Bayesian network, Support Vector Machine and Nearest Neighbor. Out of these models current study will focus on Neural Network based machine learning model.

Neural Network based Multi Layer Precepton Model (NN MLP)

There are many techniques which can be used for defect prediction such as Naïve Byes, LogReg, Random Forest, Nearest-neighbor, SVM, Machine Learning etc. In our research Feed Forward Neural Network is used for predicting the defects.

Feed forward neural networks, trained with a back-propagation learning algorithm, are the most popular neural networks. They are applied to a wide variety of problems. A FFNN consists of neurons, which are ordered into layers.

The first layer is called the input layer, the last layer is called the output layer, and the layers between are hidden layers. The used FFNN model is shown in Figure 1. Each neuron in a particular layer is connected with all neurons in the next layer.

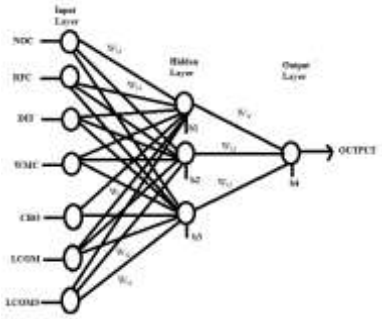

Figure 1: A Multi-layer Feed Forward Neural Network

The connection between the i^{th} and j^{th} neuron is characterized by the weight coefficient w_{ij}. The weight coefficient reflects the degree of importance of the given connection in the neural network. The output of a layer can be determined by equations

$$a=x_1w_1+x_2w_2+x_3w_3...+x_nw_n \quad (1)$$

In this research 7 neurons are used at input layer and 3 neurons are used at hidden layer. The 7 inputs are object-oriented metrics which are:-

[1] www.githu.com

[2] www.bugzila.org

NOC[12], RFC[12], DIT[12], WMC[12], CBO[12], LCOM[12], LCOM5[13].

In this research, following activation functions are used:-

1. Hyperbolic Tangent Function
2. Identity Function

Hyperbolic Tangent Function is used as activation function for hidden layer and Identity Function is used as activation function for the output layer.

Hyperbolic Tangent Function (tanh)

It takes two real-valued arguments and transforms them to a range (-1, 1).

The equation used for tanh is

$$a = \tanh(n) = \frac{2}{(1+e^{-2*n})} - 1 \qquad (2)$$

Identity Function

It takes one input and returns value n.

The equation for purelin is

$$a = Identity(n) = n \qquad (3)$$

MSE(Mean Square Error)

The MSE is the second moment of the error. Errors are calculated using Mean Square Error function as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i' - y_i\right)^2 \qquad (4)$$

where y is the actual output and y′ is the expected output.

K-means Clustering algorithm

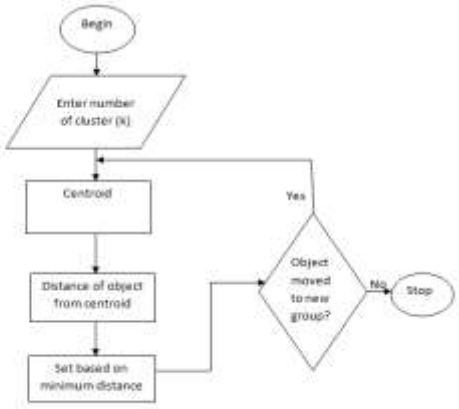

Figure 2 Flow Chart of K-means Clustering Algorithm

4. Results:-

Table 3: Results using Neural Network

	Accuracy	Precision	Recall	TNR	MSE
Licq	57.14	55.17	34.92	61.03	0.2434
Seamonkey 1.0.1	98.85	0	98.85	100	0.0258

The proposed neural network model gives accuracy up to 98%. It gives a precision rate 56% for Licq dataset. The proposed model gives a perfect TNR rate for Seamonkey dataset and 31% for Licq dataset. MSE given by proposed model is 0.2434 for Licq and 0.0258 for Seamonkey dataset.

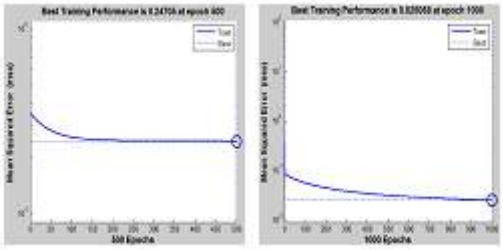

Figure 3 Training Performance (MSE) of (a) SM Version 1.0.1, (b) Licq

Results using Weka:

Cluster Centroids:

The clusters are generated using K means clustering algorithm and the following are the parameters that are used for the calculation of all the five sets i.e. Set1, Set2, Set3, Set4 and Set5:

Table 4 TP Rate

Class	Set1		Set2		Set3		Set4		Set5	
	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ
TRUE	0.468	0.603	0.489	0.611	0	0.397	0.574	0.151	0.277	0.397
FALSE	0.76	0.5	0.69	0.519	0.925	0.701	0.934	0.883	0.83	0.649
Weighted Avg.	0.757	0.546	0.688	0.561	0.914	0.564	0.93	0.554	0.823	0.536

Table 5: FP Rate

Class	Set1		Set2		Set3		Set4		Set5	
	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ
TRUE	0.24	0.5	0.31	0.481	0.075	0.299	0.066	0.117	0.17	0.351
FALSE	0.532	0.397	0.511	0.389	1	0.603	0.426	0.849	0.723	0.603
Weighted Avg.	0.529	0.443	0.508	0.43	0.989	0.466	0.421	0.52	0.717	0.49

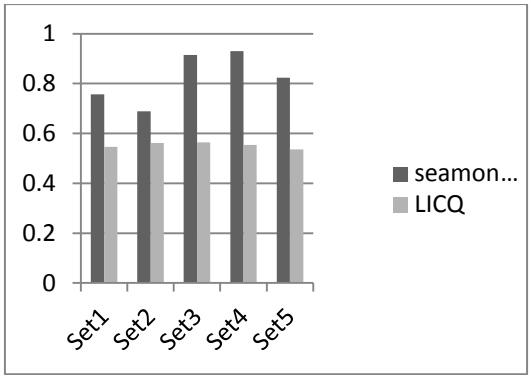

Figure 4: TP Rate using K-means Clustering

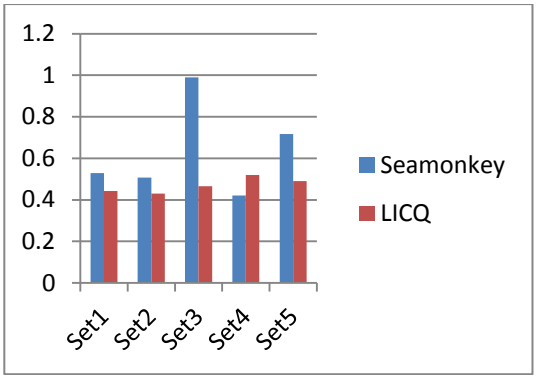

Figure 5: FP Rate using K-means Clustering

The True positive rate states the rate of faults correctly specified as faulty modules in the dataset. In seamonkey 1.0.1 dataset, the set 4 has predicted faulty modules most correctly i.e. 0.93 and the probability of being wrong is 0.07. The set 4 defines the coupling and complexity metrics, here the results define that it has the highest capability to predict the faulty modules. In LICQ dataset, set3 has provided maximum TP rate of 0.564. The set3 describes Cohesion and in case of LICQ dataset cohesion is helpful in providing good results.

The FP rate is false positive rate based on buggy and non-buggy classes. The Set3 has the highest False Positive rate i.e. 0.989.2. It is evaluated over the seamonkey 1.0.1 dataset. This data set shows that the Cohesion has the highest tendency to predict the non-faulty modules as faulty. So the cohesion must be taken care to reduce the False Positive rate and better results can be generated by the datasets. Similarly, set5 has 0.717, set1 0.529, set2 0.508 and set4 0.421 values of False positive rate .These sets predict the non-faulty modules as faulty better than the set 3 in Seamonkey 1.0.1 dataset. For LICQ dataset, 0.52 is the highest value of FP rate. It is given by set 4, which describes Coupling and Complexity.

Table 6: Precision

Class	Set1		Set2		Set3		Set4		Set5	
	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ
TRUE	0.022	0.497	0.018	0.51	0	0.521	0.092	0.514	0.019	0.481
FALSE	0.992	0.606	0.992	0.62	0.987	0.587	0.995	0.56	0.99	0.568
Weighted Avg.	0.981	0.557	0.98	0.571	0.975	0.557	0.984	0.539	0.979	0.529

Table 7: Recall

Class	Set1		Set2		Set3		Set4		Set5	
	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ
TRUE	0.468	0.603	0.489	0.611	0.489	0.397	0.574	0.151	0.277	0.397
FALSE	0.76	0.5	0.69	0.519	0.69	0.701	0.934	0.883	0.83	0.649
Weighted Avg.	0.757	0.546	0.688	0.561	0.914	0.564	0.93	0.554	0.823	0.536

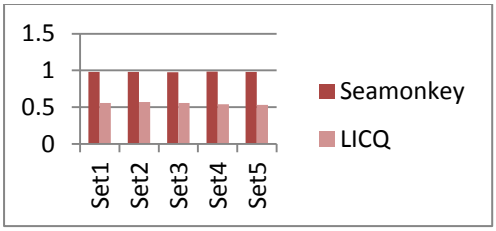

Figure 6: Precision using K-means Clustering

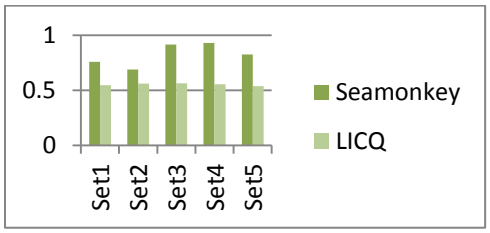

Figure 7: Recall using K-means Clustering

The precision is defined as the ratio of number of modules correctly classified to be faulty to the total number of modules. As seen in the fig. 3 the Set4 has the highest precision of 0.984 for seamonkey 1.0.1 dataset. The True positive rate of Set 4 that specifies Coupling and Complexity is maximum. Depending on True positive rate the precision is also maximum in case of the Set 4. Here it is specified that the coupling and complexity has to be taken care of, for maximizing the predicted positive cases. Encapsulation is also helpful in generating better estimations of positive cases that were correct as seen in case of Set 1. The Cohesion is most unsuccessful in giving higher value for precision in seamonkey 1.0.1 dataset. As shown in the case of Set 3.But for LICQ dataset, the results are different. In LICQ, Set2 provides the maximum value of 0.571.Set2 defines inheritance.

The maximum recall value is 0.93 which is generated by Set4 in seamonkey 1.0.1 dataset. Similarly as in the case of True positive rate and precision, set 4 predicted positive cases that were correctly identified. The coupling and complexity also provides the maximum value of recall in seamonkey 1.0.1 dataset. Recall is also known as specificity and it provides us the probability of accurate classification of a module that contains a fault. Here it has been evaluated that the set4 is able to accurately classify faulty modules. The Cohesion is capable of providing better value of recall as provided by set 3.The Inheritance is not that much successful in giving good results of recall as it is minimum for set 2 in seamonkey 1.0.1 dataset. The LICQ dataset has different results. The set 3 of Cohesion provided maximum value of 0.564 for recall. The set 5 defines Combined Object oriented properties and has the lowest value.

Table 8: ROC Area using K-means Clustering

Class	Set1		Set2		Set3		Set4		Set5	
	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ	Seamonkey	LICQ
TRUE	0.614	0.552	0.59	0.565	0.465	0.549	0.754	0.517	0.555	0.523
FALSE	0.614	0.552	0.59	0.565	0.438	0.549	0.754	0.517	0.547	0.523
Weighted Avg.	0.614	0.552	0.59	0.565	0.438	0.549	0.754	0.517	0.547	0.523

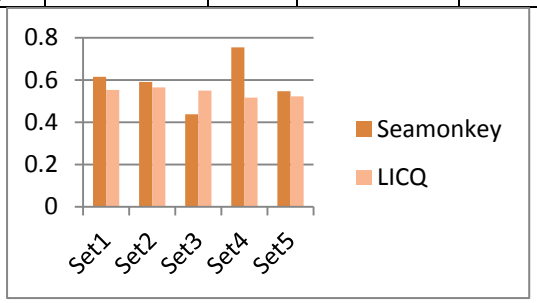

Figure 8: ROC Area using K-means Clustering

The Area under Receiver Operating Characteristics (ROC) curve provides us the relationship of True positive rate and False positive rate. ROC area is maximum for Set4 which is 0.754 in seamonkey 1.0.1 dataset. It is evaluated that the Coupling and Complexity provides us the maximum area under ROC and the Cohesion is most unsuccessful in providing the better area under ROC in seamonkey 1.0.1 dataset. The encapsulation is also useful in resulting better area under ROC as set1 has AUC of 0.614 for seamonkey 1.0.1 dataset. Unlike seamonkey dataset, for LICQ data, set 4 is failed and set2 is most successful in providing highest area under ROC. The set2 defines inheritance and it provides value 0.565.

5. CONCLUSION

After studying these results we conclude that is a good prediction system is required for predicting the software defects at an early stage of software development life cycle. Using neural network technique, the accuracy of prediction system is better than others. Neural network is based on a machine learning approach. So, it is found that machine learning models are mostly used and provides the better results. These methods can be used for cross-company projects also. The proposed model provides better accuracy and TNR. This model gives low values for MSE. The k-means algorithm is most simple and well known algorithm which can be easily implemented. For k-means clustering using Weka tool, the TP rate, precision, recall and ROC area are better. So, Weka tool is also useful in providing good results for fault and non-fault prediction.

6. REFERENCES

[1] Ramaswamy, V., Pushphavathi, T.P., and Suma, V., "Defect Prediction Apporaches for Software Projects using Genetic Fuzzy Data Mining", ICT and Critical Infrastructure, Proceedings of the 48th Annual Convention of CSI Vol II, CSI Vishakapatnam Chapter, 13-15 Dec, 2013, pp. 313-320.

[2] Rawat, M.S., Dubey, S.K., "Software Defect Prediction Models for Quality Improvement: A Literature Study", IJCSI International Journal of Computer Science Issues, Vol. 9, No 2, pp. 288-296, September, 2012.

[3] Sharma, R., Budhija, N., Singh, B., "Study of Predicting Fault Prone Software Modules" , International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, No. 2, February 2012.

[4] Rahman, F., Posnett, D. and Devanbu, P., "Recalling the "im- precision" of cross-project defect prediction," in Proceedings of the ACM-Sigsoft 20th International Symposium on the Foundations of Software Engineering (FSE-20). Research Triangle Park, NC, USA: ACM, 11-16 Nov, 2012.

[5] Fenton, N. E. and Neil, M., "A Critique of Software Defect Prediction Models", IEEE Transactions on Software Engineering, Vol. 25, No. 5, pp. 1-15, September-October 1999.

[6] Song, Q., Jia, Z., Sheppered, M., Ying, S., and Liu, J., "A General Software Defect-Proneness Prediction Framework", IEEE Transactions on Software Engineering, Vol. 37, No. 3, pp. 356-370, May-June 2011.

[7] Fenton, N. E. and Neil, M., "A Critique of Software Defect Prediction Models", IEEE Transactions on Software Engineering, Vol. 25, No. 5, pp. 1-15, September-October 1999.

[8] Ma, Y., Ma, Y., and Cukic, B., "A Statistical Framework for the Prediction of Fault-Proneness", *Advances In Machine Learning Applications In Software Engineering,* 2007.

[9] Singh, M., and Salaria, D. S., "Software Defect Prediction Tool based on Neural Network," International Journal of Computer Applications (0975 – 8887) Vol. 70, No.22, pp. 22-28, May 2013.

[10] Jain,Anil K., Dubes,Richard C.,1988 ."Algorithms for Clustering Data". Prentice Hal l. Jain ,Ani l K. , Flynn,P.,1996.Image segmentation using clustering.In :Advances in Image Understanding . IEEE Computer Society Press,pp.65 – 83
[11] Ball,G.,Hal l,D.,1965.ISO DATA,A Novel Method Of Data Analysis And Pattern Classification. Technical report NTISAD 69 9616.Stanford Research Institute, Stanford ,CA.
[12] Forgy, E .W .,1965.Cluster analysis of multivariate data : Efficiency vs .interpretability of classifications .Biometrics 21,768– 769 .
[13] Chidamber, S.R., and Kemerer, C.F., "A metrics suite for object oriented design". IEEE Transactions on Software Engineering, Vol. 20, No. 6, pp. 476–493, 1994.
[14] Henderson-Sellers, B., "Software Metrics", Prentice Hall, Hemel Hempstaed, U.K., 1996.