An Empirical Study to Evaluate the Relationship of Object-Oriented Metrics and Change Proneness

Ruchika Malhotra and Megha Khanna
Department of Computer Science and Engineering, Technological University, India

Abstract: *Software maintenance deals with changes or modifications which software goes through. Change prediction models help in identification of classes/modules which are prone to change in future releases of a software product. As change prone classes are probable sources of defects and modifications, they represent the weak areas of a product. Thus, change prediction models would aid software developers in delivering an effective quality software product by allocating more resources to change prone classes/modules as they need greater attention and resources for verification and meticulous testing. This would reduce the probability of defects in future releases and would yield a better quality product and satisfied customers. This study deals with the identification of change prone classes in an Object-Oriented (OO) software in order to evaluate whether a relationship exists between OO metrics and change proneness attribute of a class. The study also compares the effectiveness of two sets of methods for change prediction tasks i.e. the traditional statistical methods (logistic regression) and the recently widely used machine learning methods like Bagging, Multi-layer perceptron etc.*

Keywords: *Change proneness, empirical validation, machine learning, object-oriented and software quality.*

Received May 29, 2015; accepted September 20, 2015

1. Introduction

Software project planning through estimation is an important tool for researchers and practitioners [12]. It is crucial for software developers to identify change prone classes, in order to effectively distribute and manage constraint resources like effort, cost and time [7]. A class is termed as change prone, if it is likely to get modified because of a defect, future enhancement or requirement change in the forthcoming release of the software product [9, 11]. Determination of change prone classes in the early phases of software development life cycle is beneficial as developers can redesign or structure such classes in a manner which would result in minimum number of defects and changes. Also, such classes are thoroughly tested with stringent verification activities, in order to detect errors and take timely corrective actions [8]. The aim of this study is to develop prediction models which identify change prone classes of a software.

In order to develop change prediction models, it is important to evaluate the existence of relationship between object-oriented software metrics of a class and its change proneness attribute. This empirical study ascertains the change prone nature of a class based on software metrics data. The study also evaluates the effectiveness of two different categories of methods namely the traditional statistical method Logistic Regression (LR) as well as three Machine Learning (ML) methods Multilayer Perceptron (MLP), Bagging (BG) and Random Forests (RF) for evaluation of change prone classes. Another aim of the study is to identify significant metrics for change prediction in a class.

In order to analyze the stated aims, we perform an empirical study based on two open source data sets developed in Java language. The study chooses two open source data sets, as there has been a paradigm shift where open source software have been highly appreciated by both the developers because of the large community support, and the users because of its ability to customize features. Thus, it is important to examine the effectiveness of change proneness prediction models on open source data sets to support and enhance their development process and quality.

The various sections of the study are organized as follows: section 2 states the related work and section 3 presents the experimental design. Section 4 states the analysis and interpretation of results and section 5 evaluates the validity of the results. Finally, section 6 presents the conclusions and future work.

2. Related Work

Development of change prediction models is an effective way to manage change as developers can perform judicious allocation of resources and rigorous verification activities on the identified change prone classes. Lu *et al.* [9] performed statistical meta-analysis in order to evaluate the ability of 62 OO metrics for prediction of change proneness attribute. A study by Zhou *et al.* [15] concluded that there exists a confounding effect of class size on the relationship between OO metrics and change proneness. Studies by Koru and Tian [7] and Koru and Liu [8] developed tree based models to determine change prone classes. Elish and Al-Khiaty [5] advocate a combination of evolution metrics along with Chidamber and Kemerer [4] metrics suite for prediction of change prone classes.

A study by Malhotra and Khanna [11] empirically validated open source data sets to evaluate statistical as well as ML methods for ascertaining the change proneness attribute of a class. Though this study showed comparative performance of both sets of methods, it did not evaluate the comparative performance statistically. Moreover, the relationship of individual metrics was not evaluated using hypothesis. This study focuses on an effective strategy for model development using outlier analysis and effective validation strategy, which evaluates the individual relationship of each metric efficiently. The study also compares its results with a similar study performed by Malhotra and Bansal [10].

3. Experimental Design

This study explores the following Research Questions (RQ) in this study:

- *RQ1*: Is there a relationship between the object-oriented metrics and the change proneness attribute of a class?
- *RQ2*: Which metrics are significant for change prediction tasks?
- *RQ3*: What is the comparative performance of statistical and machine learning methods for development of change prediction models?

In order to answer the above research questions the experimental set up is described in the following sections.

3.1. Variable Selection

This study chooses a widely used metric suite proposed by Chidamber and Kemerer [4] as independent variables for evaluating its relationship with change proneness attribute of a class. We select this metrics suite along with Source Lines of Code (SLOC) metric as it has a representative for all software characteristics (size, coupling, cohesion and inheritance). The SLOC metric counts the number of source code lines in a particular class except the comments. The various metrics included in the CK metric suite are Lack of Cohesion amongst Methods (LCOM), Coupling Between Objects (CBO), Depth of Inheritance Tree (DIT), Number of Children (NOC), Response for a Class (RFC) and Weighted Method Complexity (WMC).The dependent variable of the study, change proneness, is binary in nature. It depicts whether a class was modified in the new release of the software or not.

3.2. Hypothesis Formulation

The study evaluates two sets of hypothesis. Hypothesis Set A analyzes the relationship of OO metrics with the change proneness attribute of a class.

Hypothesis Set B evaluates the prediction capability of statistical and ML methods for developing change prediction models.

3.2.1. Hypothesis Set A

- *Hypothesis for Cohesion Metric* (*LCOM Metric*):
  - *H0 null hypothesis*: Cohesion is not related to change proneness attribute of a class.
  - *H0 alternate hypothesis*: Cohesion has a negative impact on change proneness of a class i.e., higher the cohesion of a class, the lower would be the probability that it will change.
- *Hypothesis for coupling metric* (*CBO Metric*):
  - *H1 null hypothesis:* Coupling is not related to change proneness attribute of a class.
  - *H1 alternate hypothesis*: Coupling has a positive effect on change proneness of a class i.e., if the class is coupled to larger number of classes, the higher the probability that it will change.
- *Hypothesis for inheritance metrics* (DIT and NOC Metrics):
  - *H2 null hypothesis*: Inheritance is not related to change proneness of a class.
  - *H2 alternate hypothesis*: Inheritance has a positive effect on change proneness of a class i.e., a class with high values of inheritance metrics have higher probability of change.
- *Hypothesis for size metrics* (*SLOC*, *RFC*, *WMC Metrics*)
  - *H3 Null Hypothesis*: Size metrics are not related to change proneness attribute of a class.
  - *H3 Alternate Hypothesis*: Size has a positive effect on change proneness of a class i.e., a large class has a higher probability of change than a small class.

3.2.2. Hypothesis Set B

- Hypothesis for BG Method
  - *H4 null hypothesis*: BG does not outperform the three compared methods (LR, MLP, RF) for prediction of change prone classes.
  - *H4 alternate hypothesis*: BG outperforms the three compared methods (LR, MLP and RF) for prediction of change prone classes.

3.3. Empirical Data Collection

The two open-source data sets used in the study are Apollo and AVIsync, developed in Java programming language (http://sourceforge.net). Two versions of each software was analyzed in order to extract change statistics such as SLOC added, SLOC deleted or SLOC modified. Change in each software data set was computed from only the common classes of both the versions of a specific data set. The Apollo software is an editor and compiler for data migration software. AVISync is used for adjusting synchronization issues for DivX AVI format audio/video files. A brief description of software data sets details along with the versions analyzed, number of data points and percentage of changed classes is provided in Table 1.

The data collection procedure was same as followed by Malhotra and Khanna [11].

Table 1. Dataset details.

Software Name	Versions	No. of Data Points	Percentage of Changed Classes
Apollo	0.1-0.2	252	27%
AVISync	1.1-1.2	73	40%

3.4. Description of Methods

Univarite LR is used to assess the relationship between a specific independent variable (OO metric) and the dependent variable. This technique is used for each of the OO metrics used in the study. The evaluation of the hypothesis developed in section 3.2.1 is done on the basis of significance statistic of univariate LR results.

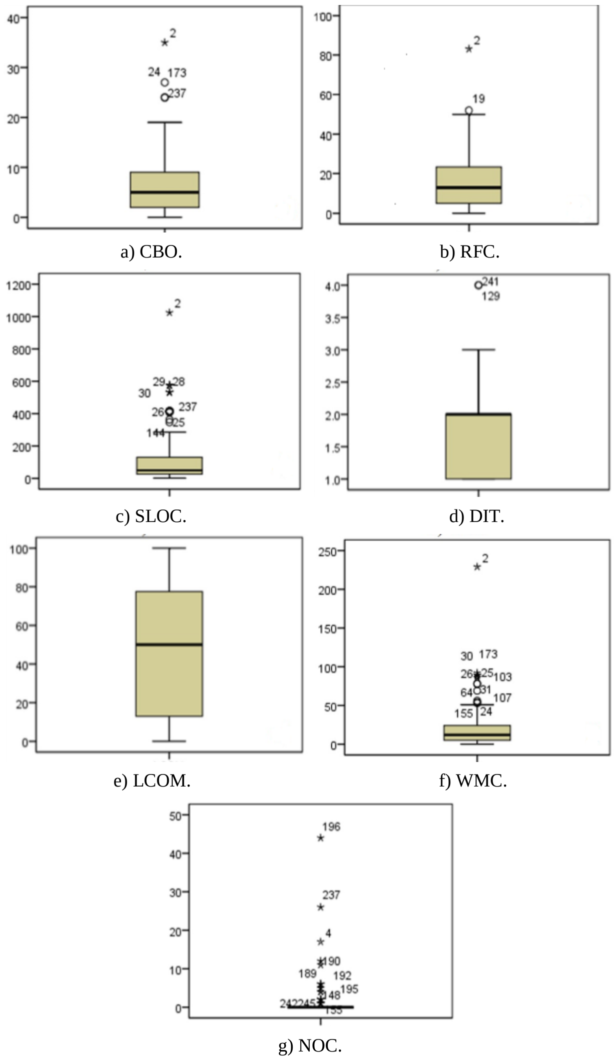
Multivariate LR uses a combination of different OO metrics in order to develop a prediction model for determining change prone classes. This study uses the aid of WEKA tool for developing change prediction models using ML methods. The study uses default parameter settings of the WEKA tool for all the methods. In order build effective change prediction models using ML methods, a feature selection technique namely Correlation based Feature Selection (CFS) was used. This technique was proposed by Hall [6].

4. Experimental Results

The aim of the study is to assess the relationship between OO metrics and change proneness. Outlier analysis was performed on the data sets obtained from each software and outliers were removed. The remaining data points were used to generate change prediction models. Apart from univariate and multivariate LR analysis, the study performed ten-fold cross validation [14] of all the models. The predicted models were evaluated on the basis of Sensitivity, Specificity, Accuracy, F-measure and Receiver Operating Characteristic (ROC) Analysis. An ROC curve represents a plot between sensitivity on the vertical axis and 1-specificity on the horizontal axis. An optimal cut off point which maximizes both the sensitivity and the specificity is selected. The effectiveness of the predicted models is assessed by evaluating Accurcay, F-measure and Area Under the ROC Curve (AUC).

4.1. Outlier Analysis

Outlier analysis is an important step for pre-processing data. According to Barnett and Lewis [1] “an outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs”. This study identifies all the outliers in each of the data set corresponding to all the metrics with the help of boxplots. Figure 1 shows the boxplots for Apollo data set for each independent variable. There were no outliers corresponding to the LCOM metric. All the data points shown outside the boxes correspond to the outlying data points.



a) CBO. b) RFC. c) SLOC. d) DIT. e) LCOM. f) WMC. g) NOC.

Figure 1. Outliers for apollo data set.

Similarly, Figure 2 shows the boxplots for AVISync data set. For the AVISync data set, there were no outliers corresponding to the RFC metric and the DIT metric.

All the outliers were removed while performing univariate and multivariate analysis and developing models for change prediction.

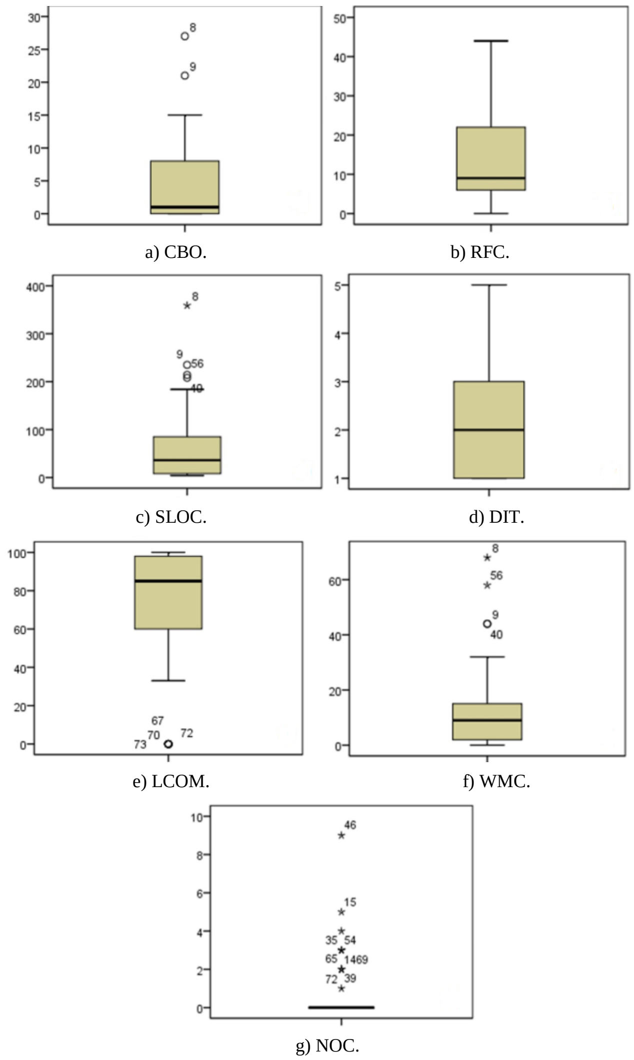


Figure 2. Outliers for AVISync data set.

4.2. Hypothesis Testing using Univariate Analysis

Table 2 provides the statistical significance for each metric in both the data sets. All the metrics with a significance value of less than 0.05 are shown in bold and are significantly related to change proneness in the corresponding data set.

For Apollo data set, four out of seven metrics were found significantly related to change proneness at a threshold level of 0.05. However, the NOC metric, the SLOC metric and the DIT metric were found insignificant on the basis of univariate analysis for Apollo data set. The univariate results of AVISync data set shows CBO, SLOC, DIT and WMC as significant metrics for a threshold value of 0.05. The metrics which were insignificant for the AVISync data set were NOC, RFC and LCOM.

Table 2. Univariate LR results.

Software name	CBO	NOC	RFC	SLOC	DIT	LCOM	WMC
Apollo	0.001	0.904	0.001	0.108	0.938	0.023	0.005
AVISync	0.012	0.405	0.093	0.008	0.009	0.597	0.002

4.3. Multivariate LR Results

A multivariate LR analysis is used to analyze the combined effect of OO metrics on the change proneness of a class. Multicollinearity depicts the extent to which the effect of a variable can be predicted by other variables in the analysis [13]. The conditional number for the models on both the data sets is below 30 indicating tolerable multicollinearity.The study uses backward elimination method for generation of multivariate LR model. Tables 3 and 4 provide the coefficient (B), standard Error (SE), statistical significance and odds ratio for the metrics which are included in the multivariate model for Apollo and AVISync data set respectively. According to Table 3, only the CBO metric was selected for inclusion in the multivariate change proneness model for Apollo data set. However, the multivariate LR model on the AVISync data set included DIT and WMC for the model development as shown in Table 4.

Table 3. Multivariate LR results of apollo data set (Backward LR).

Metric Name	B	S.E	Significance	Odds ratio
CBO	0.117	0.029	0.000	1.124
Constant	-1.706	0.241	0.000	0.182

Table 4. Multivariate LR results of AVISync data set (Backward LR).

Metric Name	B	S.E	Significance	Odds ratio
DIT	-0.478	0.255	0.061	0.620
WMC	0.083	0.030	0.007	1.086
Constant	-0.567	0.178	0.404	0.567

4.4. Ten-fold Cross Validation Results

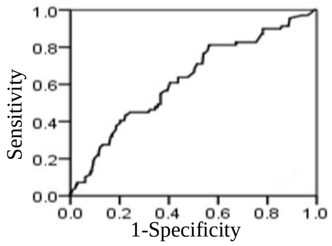
In order to minimize the features used, we first used the CFS method before application of ML methods to extract OO metrics which are highly correlated with the change in a class. The metrics which were significant with univariate analysis were selected for model prediction using the LR method. The CBO metric, the RFC metric and the SLOC metric was found useful on the Apollo data set while the CBO metric, the DIT metric and the WMC metric were found useful on the AVISync data set after applying the CFS method.

Table 5 shows the ten-fold cross validation results on the Apollo data set. According to the table, the best AUC results were given by the model developed using the BG method with a specificity and sensitivity values of 70.5% and 69.6% respectively and a cut off point of 0.245.

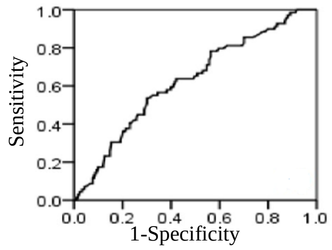
Table 5. Ten-fold cross validation results of apollo dataset.

Method	Specificity	Sensitivity	Cut-off Point	AUC	Accuracy	F-measure
LR	59.6 %	60.9 %	0.247	0.626	73.1%	66.5%
MLP	59.6 %	59.4 %	0.291	0.627	71.8%	63.3%
BG	70.5 %	69.6 %	0.245	0.766	71.8%	68.8%
RF	71.6 %	68.1 %	0.212	0.758	72.6%	72.0%

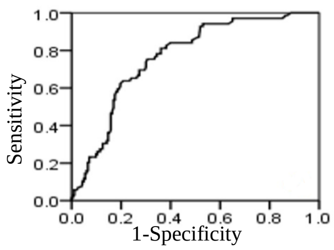
The AUC value was computed as 0.766, with good accuracy value of 71.8% and F-measure value of 68.8% for the BG model. The model developed by the RF method also showed good results with an AUC value of 0.758, an accuracy of 72.6% and best F-measure value of 72.0%. The models developed by the LR method and the MLP method were comparable with an AUC value of 0.626 and 0.627, accuracy values of 73.1% and 71.8% and F-measure of 66.5% and 63.3% respectively. The ROC curves of the generated models on Apollo data set are shown in Figure 3.



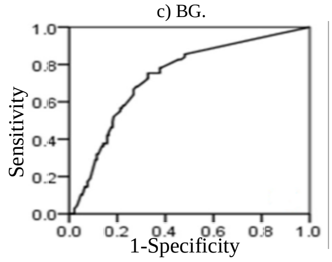
a) LR.



b) MLP.



c) BG.



d) RF.

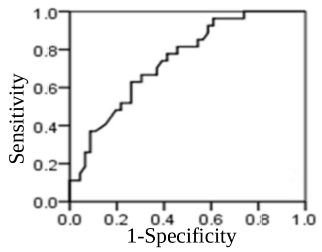
Figure 3. ROC for apollo.

Table 6 states the ten-fold cross validation results of the AVISync data set. The best AUC results were given by the model developed using the MLP method with an AUC value of 0.784, but it gave low accuracy and F-measure value of 71.2% and 69.1% respectively.

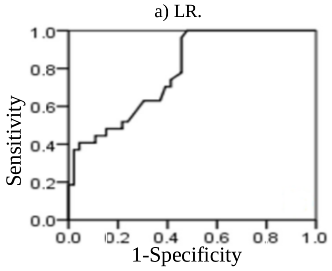
Table 6. Ten-fold cross validation results of AVISync dataset.

Method	Specificity	Sensitivity	Cut-off Point	AUC	Accuracy	F-measure
LR	67.4 %	66.7 %	0.363	0.743	67.1%	66.5%
MLP	63.0 %	63.0 %	0.405	0.784	71.2%	69.1%
BG	65.2 %	66.7 %	0.401	0.759	73.9%	75.9%
RF	63.0 %	63.0 %	0.344	0.745	75.3%	79.6%

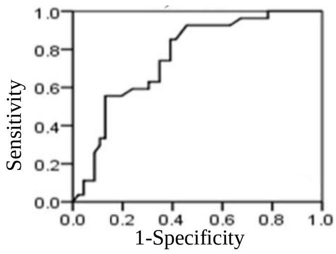
The next best AUC value was given by the BG model as 0.759 with accuracy value of 73.9% and F-measure value of 75.9%. The BG model gave a specificity and sensitivity value of 65.2% and 66.7% respectively at a cut-off point of 0.401. The AUC's of the model developed using the RF method and the LR method were 0.745 and 0.743 respectively. The accuracy and F-measure values of the RF method were good with 75.3% and 79.6% respectively. The ROC curves of the change prediction models developed using different methods on the AVISync data set are shown in Figure 4.



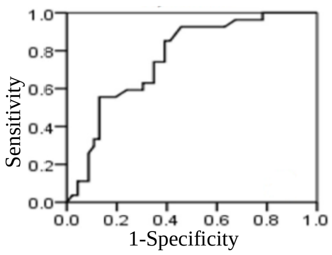
a) LR.



b) MLP.



c) BG.



d) RF.

Figure 4. ROC for AVISync.

5. Discussion of Results

This section presents a detailed discussion of all experimental results.

5.1. Hypothesis Evaluation for OO Metrics

Hypothesis set A (Section 3.2.1) is evaluated on the basis of univariate results discussed in Section 4.3. The results are compared with a study performed on change proneness prediction by Malhotra and Bansal [10].

- *Hypothesis for cohesion metric*: The cohesion metric LCOM was found to be significant in the Apollo data set. However, the LCOM metric was found insignificant in the AVISync data set. The metric was found to be a significant predictor of change proneness in a study by Malhotra and Bansal [10]. Thus we reject the Null Hypothesis H0.
- *Hypothesis for coupling metric*: The coupling metric CBO was found to be significant in both the data sets used in the study. Similar results were shown by Malhotra and Bansal [10]. Thus, we reject the Null Hypothesis H1.
- *Hypothesis for inheritance metrics*: The inheritance metric NOC was found insignificant while performing univariate LR results on both the data sets. However, the DIT metric yielded as a significant metric in the AVISync data set but insignificant in the Apollo data set. Malhotra and Bansal [10] found both DIT and NOC metrics as insignificant while predicting change proneness of an OO software. Thus, we accept Null Hypothesis H2.
- *Hypothesis for size metrics*: The study evaluated three size metrics (SLOC, RFC and WMC). The univariate results on the Apollo data set yielded RFC and WMC as significant metrics for predicting change proneness attribute. While performing univariate analysis on the AVISync data set, the SLOC and WMC metric were predicted as significant. Moreover, the study by Malhotra and Bansal [10] advocate the RFC and SLOC metrics as significant predictors of change proneness. Thus, we accept the Alternate Hypothesis H3.

5.2. Model Evaluation

Hypothesis Set B (Section 3.2.2) is evaluated on the basis of ten-fold cross validation results discussed in Section 4.5. This hypothesis tests the effectiveness of the BG method for prediction of change prone classes as compared to all the other methods used in the study.

As discussed in Section 4.5, the results of the model developed using the BG method were better than the models developed using all the other methods used in the study for the Apollo data set. The BG method gave good accuracy, F-measure and AUC values. The model developed using the BG method also showed very good results on the AVISync data set. These results were better than the models developed using the LR and RF methods on the AVISync data set as LR showed low values for AUC, accuracy and F-measure. Also though RF showed good accuracy and F-measure values, the AUC was comparatively less than that of the BG model on AVISync data set.

In order to test the significance of the results produced by the model developed using the BG method, paired t-tail test was used. The test evaluated the whether the results of the BG method were significant when compared with the models developed using the other methods by evaluating the AUC value across each validation fold of both the data sets. The results showed that the BG method was significantly better than the LR method for change proneness prediction on both the data sets at a threshold value of 0.05. Although the results of the models developed using the BG method were better than both the MLP models and the RF models but they did not significantly outperform the MLP and RF models.

5.3. Discussion

Hypothesis Set A evaluates the OO metrics which are highly correlated with change in a class. These metrics can be used in the initial phases of software development life cycle to identify change prone classes. The results of the study indicate that coupling, cohesion and size metrics are efficient indicators of change in a class. Thus, researchers and practitioners can use these metrics as indicators of change. Moreover, they can form quality benchmarks by controlling these attributes of a class to a desired level in order to prevent defects and changes in classes.

Hypothesis Set B is used to evaluate different methods in order to compare their capability to develop change prediction models. The results indicate the BG method as the best performing method for model development. These results were significantly better than the traditional method LR. Moreover, other ML algorithms, RF and MLP also gave good results when used for developing change prediction models. Thus, ML models can be efficiently used for creating change prediction models in an OO paradigm.

6. Validity Evaluation

The two major areas of validity concern for this study are construct validity and external validity.

Construct validity refers to the extent of accurate representation of the concepts measured by the dependent and the independent variables [15]. The study uses a number of OO metrics which represents concepts like coupling, cohesion, inheritance etc., Previous studies [2-3] have already investigated the accuracies of these measures.

External validity indicates the extent to which the results of the study can be generalized. This threat can be minimized by analyzing the results of the study on different data sets. In order to reduce the threat to external validity, the study used two medium sized data sets developed in Java language. However, the universal application of the results of the study cannot be claimed on all kinds of software.

7. Conclusions and Future Work

The aim of the study was to analyze the relationship between OO metrics and change proneness attribute of a class in an OO software and identify OO metrics which are good predictors of change proneness. The study also evaluates the capability of statistical and ML methods for prediction of change prone classes. The study performed empirical validation on two open source data sets developed in Java language and analyzed the results of the predicted models using ROC analysis, accuracy and F-measure. The main results of the study can be summarized as follows:

- *RQ1*: The study ascertains the existence of the relationship between OO metrics and change proneness of a class. This relationship can be utilized to predict models for determining change prone classes. The predicted results can help the software industry in efficient allocation of limited resources during maintenance and testing phases. It would also help practitioners in developing good quality software products at optimum costs as changes and defects can be identified in the early phases of software development life cycle.
- *RQ2*: The coupling metric (CBO), cohesion metric (LCOM) and size metrics (SLOC, RFC, WMC) were significant predictors of change proneness attribute of a class as they were found significant using univariate LR analysis. They were selected by the backward elimination method to be included in the model for multivariate LR. The metrics were also chosen by the CFS method for developing change prediction models using ML techniques. However, the inheritance metrics (DIT and NOC) were found insignificant while developing change proneness prediction models.
- *RQ3*: The results of the prediction model developed using the BG method outperformed the prediction results of the models developed by all the other methods (LR, MLP and RF) in the study. Though all the methods employed in the study showed good AUC, accuracy and F-measure values for the predicted models, the results of the study indicate that the ML methods gave better results for developing change prediction models as compared to the traditional statistical method LR. Thus, the ML methods showed competitive results with the LR method. Researchers and practitioners may use ML methods for identifying change prone classes.

Future studies may empirically validate software data sets with different characteristics such as different programming language and development environment for developing change prediction models. Researchers may also evaluate a number of other ML methods to compare and assess their effectiveness for change prediction tasks. Moreover, a new class of methods namely search based algorithms may be assessed for determining change prone classes.

References

[1] Barnett V. and Lewis T., *Outliers in Statistical Data*, John Wiley, 1994.

[2] Briand L., Daly J., and Wust J., "A Unified Framework for Cohesion Measurement in Object-Oriented Systems," *Empirical Software Engineering*, vol. 3, no. 1, pp. 65-117, 1998.

[3] Briand J., Daly J., and Wust J., "A Unified Framework for Coupling Measurement in Object-Oriented Systems," *IEEE Transactions on Software Engineering*, vol. 25, no. 1, pp. 91-121, 1999.

[4] Chidamber S. and Kemerer C., "A Metrics Suite for Object-Oriented Design," *IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476-493, 1994.

[5] Elish M. and Al-Khiaty M., "A Suite for Quantifying Historical Changes to Predict Future Change-prone Classes in Object-Oriented Software," *Journal of Software: Evolution and Process*, vol. 25, no.5, pp. 407-437, 2013.

[6] Hall M., "Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning," *in Proceedings of 17th International Conference on Machine Learning*, CA, pp. 359-366, 2000.

[7] Koru A. and Tian J., "Comparing High-Change Modules and Modules with the Highest Measurement Valuesin Two Large-Scale Open-Source Products," *IEEE Transactions on Software Engineering*, vol. 31, no. 8, pp. 625-642, 2005.

[8] Koru A. and Liu H., "Identifying and Characterizing Change-Prone Classes in Two Large-Scale Open-Source Products," *Journal of Systems and Software*, vol. 80, no. 1, pp. 63-73, 2007.

[9] Lu H., Zhou Y., Xu B., Leung H., and Chen L., "The Ability of Object-Oriented Metrics to Predict Change Proneness: A Meta-Analysis," *Empirical Software Engineering*, vol. 17, no. 3, pp. 200-242, 2012.

[10] Malhotra R. and Bansal A., "Predicting Software Change in an Open Source Software Using Machine Learning Algorithms," *International Journal of Reliability, Quality and Safety Engineering*, vol. 20, no. 6, 2014.

[11] Malhotra R. and Khanna M., "Investigation of Relationship between Object-Oriented Metrics and Change Proneness," *International Journal of Machine Learning and Cybernetics*, vol. 4, no. 4, pp. 273-286, 2013.

[12] Nagpal G., Uddin M., and Kaur A., "Grey Relational Effort Analysis Technique using Regression Methods for Software Effort

Estimation," *The International Arab Journal of Information Technology*, vol. 11, no. 5, pp. 437-446, 2014.

[13] Singh Y., Kaur A., and Malhotra R., "Empirical Validation of Object-Oriented Metrics for Predicting Fault Proneness Models," *Software Quality Journal*, vol. 18, no. 1, pp. 3-35, 2010.

[14] Stone M., "Cross-Validatory Choice and Assessment of Statistical Predictions," *Royal Society Series B*, vol. 36, no. 2, pp. 111-114, 1974.

[15] Zhou Y., Leung H., and Xu B., "Examining the Potentially Confounding Effect of Class Size on the Associations Between Object-Oriented Metrics and Change Proneness," *IEEE Transactions on Software Engineering*, vol. 35, no. 5, pp. 607-623, 2009.

Ruchika Malhotra is Associate Head and Associate Professor in the Discipline of Software Engineering, Department of Computer Science & Engineering, Delhi Technological University (formerly Delhi College of Engineering), Delhi, India. She was awarded with prestigious Raman Fellowship for pursuing Post doctoral research in Indiana University Purdue University Indianapolis USA. She received her master's and doctorate degree in software engineering from the University School of Information Technology, Guru Gobind Singh Indraprastha University, Delhi, India. She was an Assistant Professor at the University School of Information Technology, Guru Gobind Singh Indraprastha University, Delhi, India. She has received IBM Faculty Award 2013. She is recipient of Commendable Research Award by Delhi Technological University. Her h-index is 24 as reported by Google Scholar. She is author of book titled "Empirical Research in Software Engineering" published by CRC press and co-author of a book on Object Oriented Software Engineering published by PHI Learning. Her research interests are in software testing, improving software quality, statistical and adaptive prediction models, software metrics and the definition and validation of software metrics. She has published more than 150 research papers in international journals and conferences.

Megha Khanna is currently pursuing her doctoral degree from Delhi Technological University. She is currently working as Assistant Professor in Sri Guru Gobind Singh College of Commerce, University of Delhi. She completed her master's degree in software engineering in 2010 from the University School of Information Technology, Guru Gobind Singh Indraprastha University, India. She received her graduation degree in computer science (Hons.) in 2007 from Acharya Narendra Dev College, University of Delhi. Her research interests are in software quality improvement, applications of machine learning techniques in change prediction, and the definition and validation of software metrics. She has various publications in international conferences and journals.