FP- International Journal of Computer Science Research (IJCSR)
Volume 2, Issue 2, Pages 126-131, June 2015

PCA Based Improved Feature Selection Using Software Metrics Correlation Data for Improved Decision Making

Shweta
Department of Computer Science
Chandigarh University
Gharuan,Punab
sainishweta98@gmail.com

Sugandha
Sharma Department of Computer Science Chandigarh UniversityGharuan,Punab
sugandha.ss1@gmail.com

Rupinder Singh
Department of Computer Science
Chandigarh University
Gharuan,Punab
rupipanjagotra1989@gmail.com

ABSTRACT
A large number of metrics are available in the market today so it's difficult to choose the relevant metrics for a particular context and further to control them. Thus, correlation is mandatory to be calculated between various metrics. PCA based feature selection is applied on the correlated metrics data for finding the most significant features. These features can be used for improving decision making by making available the principal features, reducing the use of redundant metrics.

Keywords
Software metrics, complexity, correlation, feature selection.

1. Introduction
Decision making in software industry is a continuous process and affects almost all segments of the Software Development Life Cycle (SDLC) and even after development it affects the software in the terms of maintainability and quality of service. Decision making can include decisions regarding the issue of funds, deployment of manpower for developing– team, testing, manpower regularisation and many others. While economic decisions are taken on the basis of software cost estimation methods but the decisions regarding regularisation of software testing team depends a lot on the number of reported bugs and fault prediction. Measure like maintainability, reusability, efficiency, usability etc. are known as the external metrics which need to be predicted in advance for optimal decision making process[1]. Over recent years, it has been found that all these factors depend a lot on some of the factors or parameters of the software, more commonly called software internal metrics. Basically, the internal metrics are those that can be directly measured and the external metrics are the ones we are interested in. Various researches have been conducted on finding the influence of these internal metrics on the external metrics. For example, a higher number of code lines leads to greater software complexity. Due to the appearance of certain factors affecting the software quality, many researchers believe that there is a direct relationship between internal attributes like-cost, effort, LOC, speed and the external software product attributes like: quality, functionality, maintainability, reliability, efficiency or complexity. [4] A strong correlation is found to exist between internal and external software metrics also. In this paper a new technique of finding correlation between metrics is using PCA based feature selection is proposed and experimentally validated for improving the decision making process.

The outline of this paper is as following: Section 2 presents some of the important metrics used in this experiment. Section 3 describes the need of finding correlation between software metrics. Section 4 describes the experiment and its findings. Section 5 includes the conclusion and future scope.

2. The internal and external metrics

Software metrics have a great role in measuring different quality aspects of the software. [6]The internal metrics are the ones that can be directly measured from the source code ex- LOC, Mc CabeCyclomatic Complexity and Halstead Volume etc.[2] The external metrics are the attributes of the software which are dependent on the values of the internal software metrics.[3] For example- reliability, maintainability, usability, efficiency, reusability, fault prediction etc. The external metric used in this experiment is the fault prediction.

3. Reason behind finding the correlation between metrics

A software product's behaviour is influenced by the internal attributes like-cost, effort, LOC and the external metrics like-reliability, maintainability, usability and fault proneness etc. and the relation between them. The software metrics are a combination of these attributes. A large number of software metrics are used to manage and control the software product. As the number of metrics applied on software increases, its management and control also increases. This can be reduced by utilizing the correlation between the software metrics. This can be illustrated with an example.

Suppose, there are two metrics 'A' and 'B'. Let the metric 'A' be used to measure the complexity so as the complexity increases, the value of 'A' also increases. Let metric 'B' be used to measure the reusability so as the reusability increases, the value of 'B' also increases.The aim is to minimize the value of 'B' that will minimize the complexity and to maximize the value of 'A' that will maximize the reusability. If we study this relation between the two metrics we find that if value of 'A' increases, the value of 'B' also increases.So, from the correlation between two metrics we can conclude that the metric 'A' can be used to assess the reusability in addition to complexity. Similarly the metric 'B' can be used to assess the complexity in addition to reusability [4].As such both the metrics can act as validation for each other and can also replace each other in some contexts. This makes the use of two metrics together, more managed and controlled.

Figure 1.1 Relations between Metrics A and B

On similar basis, the correlation between different software metrics can be used to control the software development, testing and maintenance in an improved way.

4. PCA based approach for finding the correlation

It is revealed from the literature that the values of internal metrics directly affect the value of external metrics. In this experiment the dataset used for different metrics is PROMISE data repository of NASA. The correlation among the internal metrics is found using PCA in MATLAB and linear regression line is drawn for inference.

Figure 4.1 describes the correlation between Mc Cabes LOC and Halstead Volume. The correlation coefficient is calculated to be positive and is equal to 0.90029. The scatter aggregation along the regression line shows that the two metrics are highly correlated to each other.

Figure 4.2 shows the cross correlation between Mc Cabes LOC and Halstead Volume. The steepness of the curve shows that the two metrics are highly correlated with each other.

Figure 4.3 describes the correlation between the Mc CabesCyclomatic Complexity and Halstead-Volume. The correlation coefficient is found to be positive and is equal to 0.686 showing the positive correlation between the metrics. The aggregation of the scatters along the regression line also shows that the correlation is high.

Figure 4.4 describes the cross corelation between Cyclomatic Complexity and halstead Volume. The steepness of the curve shows that the correlation is high among both the metrics.

Figure 4.5 describes the correlation between Mc Cabes LOC and Cyclomatic Complexity. The correlation coefficient is found to be 0.81776 which shows the positive correlation between the two metrics.

Figure 4.6 shows the cross correlation between Mc Cabes LOC and Cyclomatic Complexity. The steepness of the curve shows that the correlation is high.

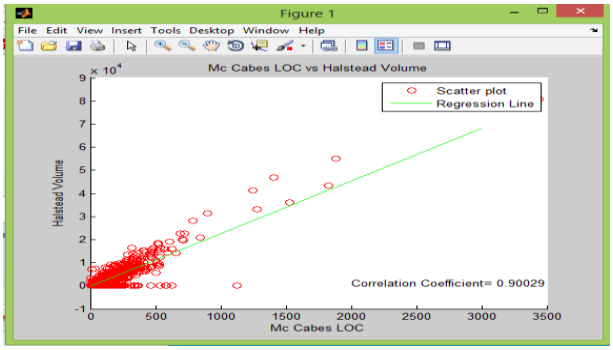

Figure 4.1: Correlation between Mc Cabes LOC and Halstead Volume

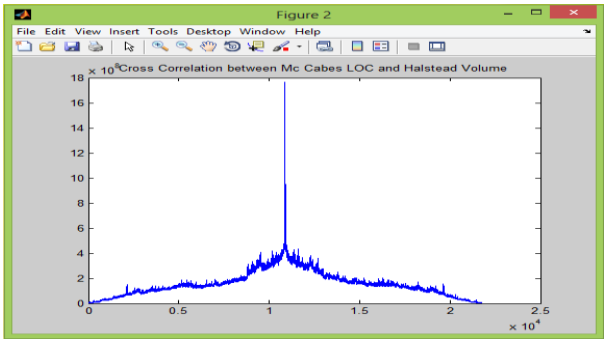

Figure 4.2: Cross Correlation between Mc Cabes LOC and Halstead Volume

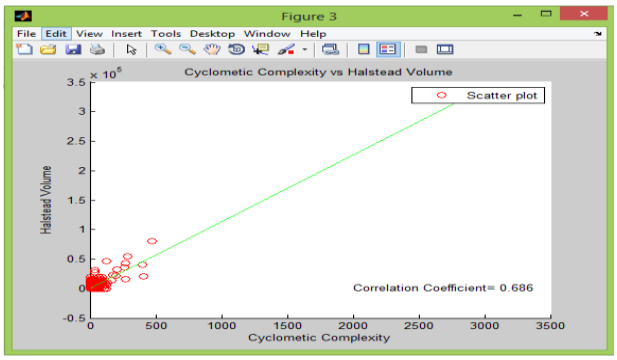

Figure 4.3: Correlation between Cyclomatic Complexity and Halstead Volume.

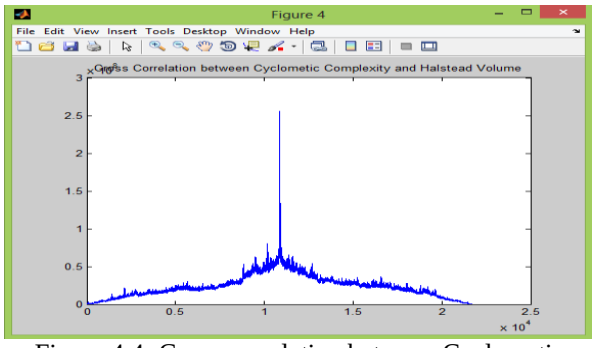

Figure 4.4: Cross correlation between Cyclomatic Complexity and halstead Volume.

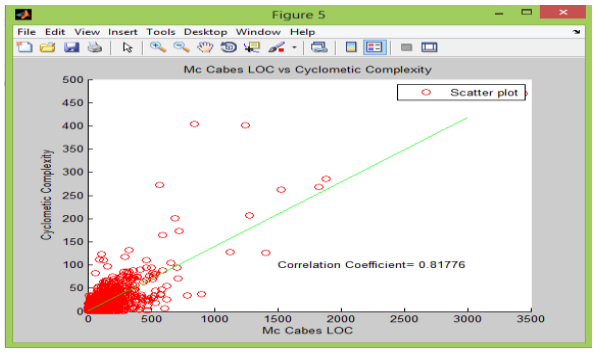

Figure 4.5: Correlation between Mc Cabes LOC and Cyclomatic Complexity.

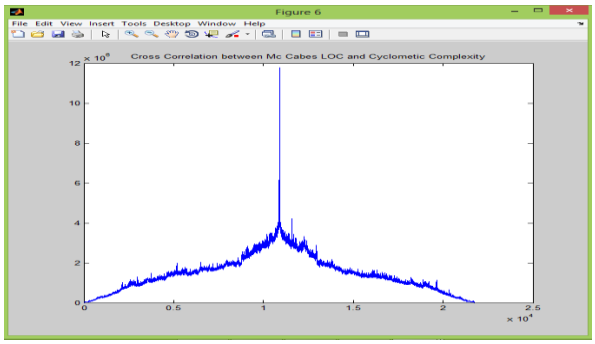

Figure 4.6: Cross correlation between Mc Cabes LOC and Cyclomatic Complexity

The next section of implementation includes the correlation between various internal metrics and the external metric: fault data. Some of the important findings are:

Figure 4.7 shows the correlation between Mc Cabes LOC and the fault data. The correlation coefficient is 0.8384 which shows positive correlation. Figure 4.8 describes the cross correlation between two metrics. The steepness of the curve shows high correlation.

Figure 4.9 shows the correlation between Mc CabesCyclomatic Complexity and the fault data. The correlation coefficient is 0.97191 which shows positive correlation between the two metrics. The correlation coefficient is higher than that of Mc Cabes LOC with fault data, that implies complexity has more effect on fault occurrence than the LOC.

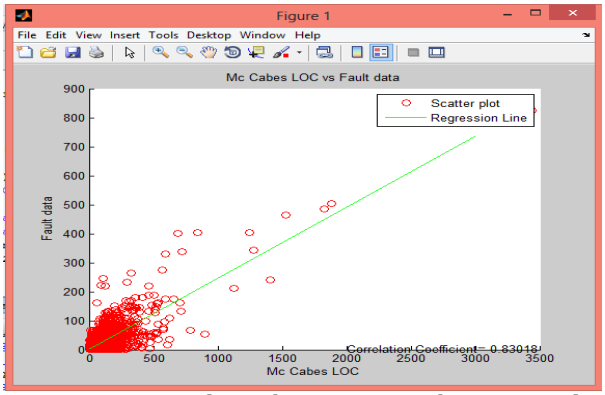

Figure 4.7: Correlation between Mc Cabes LOC and Fault Data.

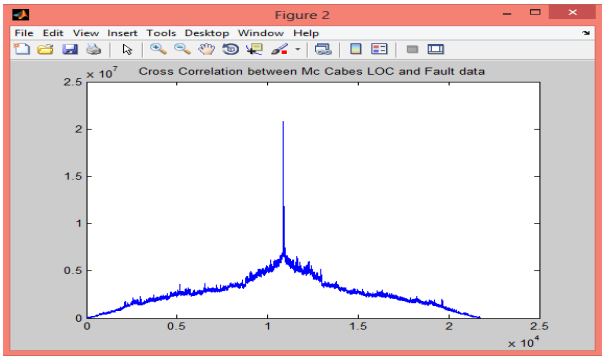

Figure 4.8: Cross Correlation between Mc Cabes LOC and Fault Data.

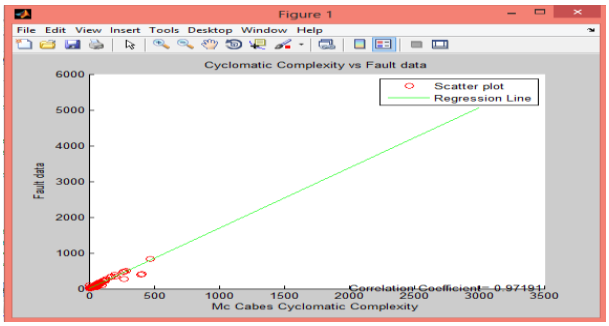

Figure 4.9: Correlation between Mc CabesCyclomatic Complexity and Fault data.

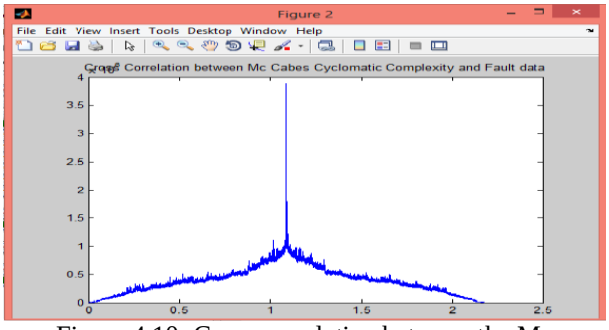

Figure 4.10: Cross correlation between the Mc CabesCyclomatic Complexity and Fault data.

Figure 4.11, 4.12 shows the positive correlation between Mc Cabes Design Complexity and fault data.

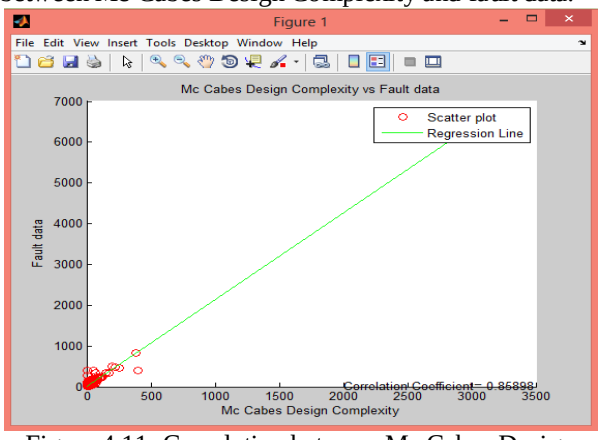

Figure 4.11: Correlation between Mc Cabes Design Complexity and fault data.

The correlation coefficient is 0.85890 which is smaller than that of the Mc CabesCyclomatic Complexity. As such the Halstead Volume has lesser impact on defect occurrence as compared to Cyclomatic Complexity.

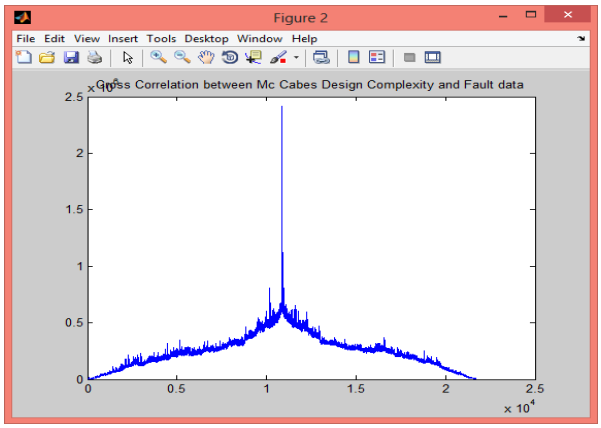

Figure 4.12: Cross Correlation between Mc Cabes Design Complexity and fault data.

Figure 4.13, 4.14 shows the positive correlation between Halstead Volume and Fault data.

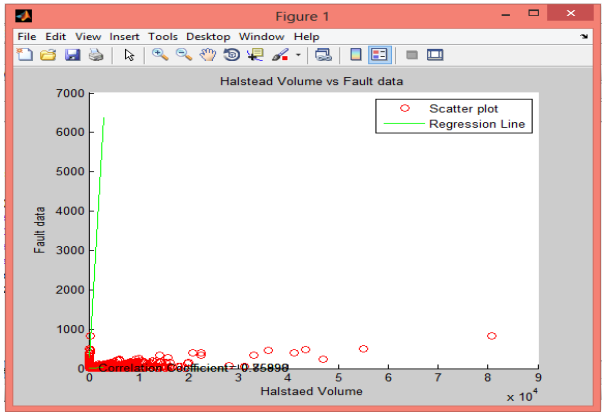

Figure 4.13: Correlation between Halstead Volume and Fault data.

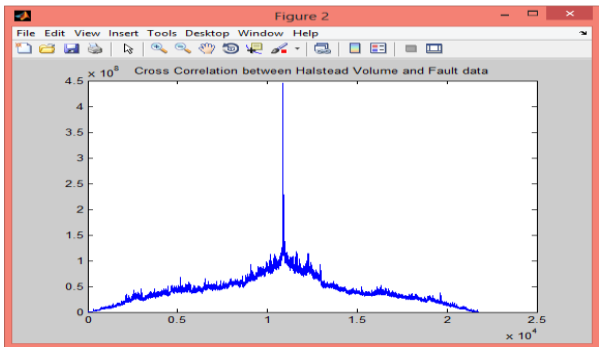

Figure 4.14: Cross Correlation between Halstead Volume and Fault data.

Figure 4.15, 4.16 shows the correlation between Halstead Difficulty and fault data. The correlation coefficient is 0.67197 which is smaller than that of the Halstead Volume, so it has lesser impact on the fault occurrence than the Halstead Volume.

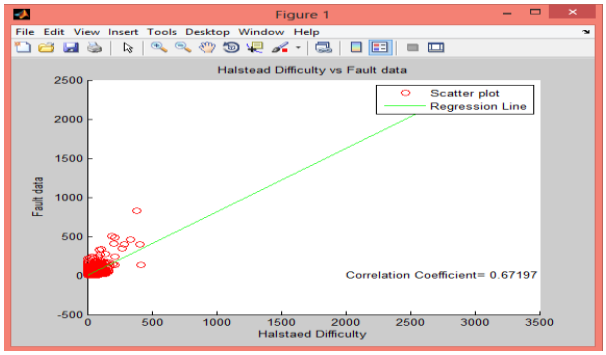

Figure 4.15: correlation between Halstead Difficulty and fault data.

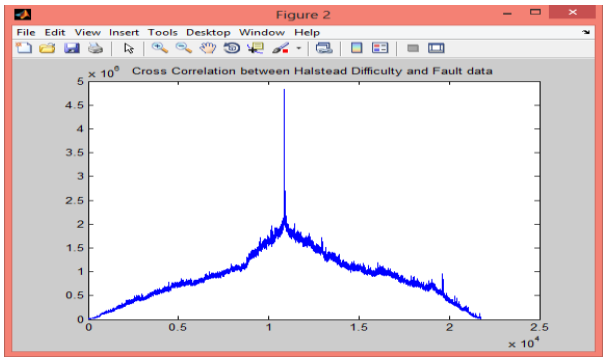

Figure: 4.16: Cross correlation between Halstead Difficulty and fault data.

The next step of the implementation is applying feature selection using Principal Component Analysis (PCA). Feature selection is used to select the most significant features among the available.[10] PCA is an orthogonal transformation that converts the set of correlated values into uncorrelated variables termed as 'principal components'. Transformation is done in such a way that the first principal component has the largest variance possible and every succeeding variable has the highest

variance possible uncorrelated to the preceding component[11].

After applying the PCA the metrics data set that contained 21 metrics data initially is reduced into the data set of 6 principal metrics.
Figure 4.17 shows the variance in the reduced data set .

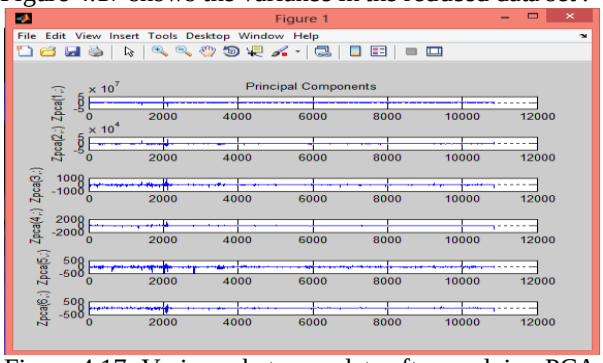

Figure 4.17: Variance between data after applying PCA

The next step includes finding the correlation of the internal metrics with the fault data. For it the last label of fault data is added to the new reduced data set to make the correlation possible. Figure 4.18 shows the correlation between PCA reduced metric data and the external measure: fault data. Figure 4.19 shows the cross correlation between the PCA reduced data and the external measure: fault data.

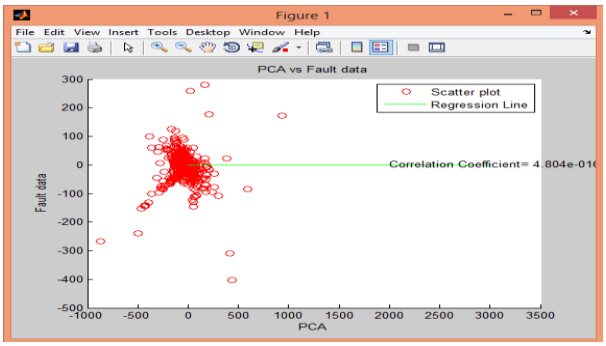

Figure 4.18: Correlation between PCA reduced data and fault data

Figure 4.18 shows the correlation between PCA reduced metric data and the external measure: fault data. Figure 4.19 shows the cross correlation between the PCA reduced data and the external measure: fault data.

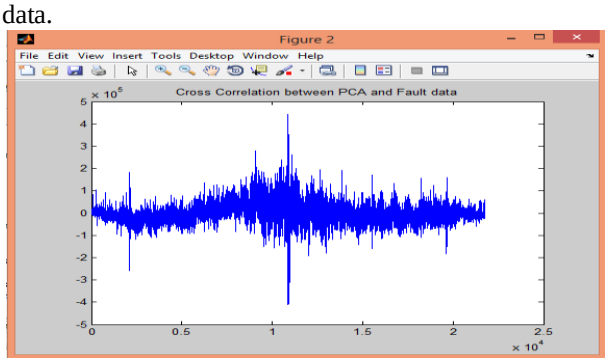

Figure 4.19: Cross Correlation between PCA reduced data and fault data

The positive correlation of the fault data with PCA reduced data signifies that the reduced set also contains the correlation between the internal and external metrics. The new found correlation values and cross-correlation functions can be utilised by the managers and business analysts as new indictors of fault prediction and their decision making process can be improved by using these results. This is a dynamic process and is routinely monitored and has the edge over traditional decision making process as it includes a scientific approach by involving statistical data.

5. Conclusion and future work

The correlation is found among the internal software metrics and between the external software metrics. Many internal metrics like: - Mc Cabes LOC, Mc CabesCyclomatic Complexity and Halstead metrics are found to be highly correlated with each other. Positive correlation is also found between these internal metrics and the external metric of fault data. After the correlation been found in the dataset, it is reduced using PCA which is also further found to be correlated with fault data. The new found correlated values and cross correlation functions can be used by the project managers and business analysts as new indicators of fault prediction. These results can be used in improving the decision making process.

The proposed technique has focused on improving the decision making by using the correlation between metrics. But in future there are still some points that need to be explored further. The proposed technique can be enhanced by using it for other external metrics also in addition to the fault prediction which will aid decision making process and other processes of the Software Development Life Cycle (SDLC). The technique can be further explored by using the metrics other than the considered and by using bigger datasets for more precision.

6. REFERENCES

[1] Mathur, Kirti, and Amber Jain. "A Comparative Survey of Software Quality Metrics."*International Journal of Research*(2013),Issue-4,Volume.2

[2] E. Fenton and Martin Neil, Software Metrics: Roadmap, International Conference on Software Engineering, Limerick, Ireland, pp 357-370, 2000,

[3] Honglei, Tu, Sun Wei, and Zhang Yanan. "The research on software metrics and software complexity metrics."*Computer Science-Technology and Applications, 2009.IFCSTA'09.International Forum on*.Vol. 1. IEEE, 2009

[4] Tashtoush, Yahva, Mohammed Al-Maolegi, and Bassam Arkok. "The correlation among Software Complexity Metrics

with Case Study."*International Journal of Advanced Computer Research*:Issue-15,Vol.4 (2014).

[5] Senousy, Mohamed B., and Tamer ShMazen. "Correlations and Weights of Maintainability Index (MI) of Open source Linux Kernel Modules." *International Journal of Computer Applications* 91.7 (2014):30-37.

[6] Rawat, Mrinal Singh, Arpita Mittal, and Sanjay Kumar Dubey. "Survey on Impact of Software Metrics on Software Quality." *International Journal of Advanced Computer Science & Applications* 3.1 (2012).

[7] Debbarma, MrinalKanti, et al. "A Review and Analysis of Software Complexity Metrics in Structural Testing." *International Journal of Computer and Communication Engineering* 2 (2013): 129-133.

[8] Pressman, Roger S. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.

[9] Gao, Kehan, et.al "Choosing software metrics for defect prediction: An investigation on feature selection techniques." *Software: Practice and Experience* 41.5(2011):579-606.

[10] Wang, Huanjing, et.al "Measuring robustness of feature selection techniques on software engineering datasets." Information Reuse and Integration (IRI), 2011 IEEE *International Conference on* IEEE,2011.

[11] Bro, Rasmus, and Age K. Smilde. "Principal component analysis."*Analytical Methods* 6.9 (2014): 2812-2831.