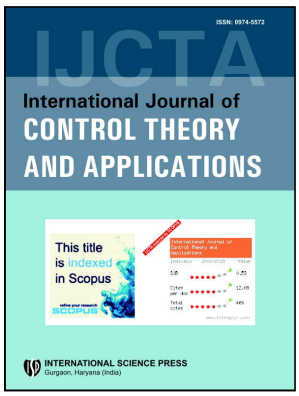

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 15 • 2017

Comparison of Machine learning tools for Defect Prediction

Deepti Chopra[1], Priyanka Dhiman[1] and Priya Dhiman[1]

[1] Department of Computer Science Indraprastha College for Women, University of Delhi
Delhi, India, Emails: dchopra27@gmail.com, prdhiman356@gmail.com, pdhiman356@gmail.com

Abstract: Defect prediction is a growing research area in the field of software engineering. Defect prediction is required for maintenance of the quality of the software. Finding defects helps in reducing the development cost of software. In this study, defect prediction is done by calculating the History Complexity metrics (HCM) for a subsystem of Eclipse. Eclipse provides an integrated development environment (IDE) for developing java based applications. A comparison was made between results that are obtained for Simple Linear Regression (SLR) and Support Vector Regression (SVR) using two different tools, namely Weka and RapidMiner. Performance of RapidMiner was found to be better for both the techniques.

Keywords: Eclipse; defect prediction; RapidMiner; Weka

1. INTRODUCTION

Data mining is the practice of searching a large amount of data in order to reveal new and relevant information. It includes data preparation and data modeling. Rapidminer, Weka , KNIME and Apache Mahout are most common data mining tools used. In our paper, we have shown the comparison between Weka and RapidMiner using two techniques: Simple Linear Regression (SLR) and Support Vector Regression (SVR). Weka is an open source tool that comprises of a collection of algorithms that can be easily applied to datasets [1]. It provides techniques for regression, data preprocessing and classification [2]. Rapidminer gives an integrated environment for Data analytics, data mining, and text mining. Using this tool, the users can easily perform various analyses such as Regression analysis, Gaussian process, and other statistical processes.

A software metric is used for measuring the degree to which a certain characteristic is possessed by a software system. Software metrics are applicable in many fields such as software debugging, cost estimation and software performance optimization. Defect prediction is one of the key research areas, which allows software developers to improve the quality of the software.

In their paper, Singh and Chaturvedi [3] applied Support Vector regression and analyzed the bug occurrence based on the complexity of code changes. Their study reveals that SVR models are more applicable for predicting

the future occurrence of bugs. In his paper, Hassan [4] presented a defect prediction metric to predict the future bugs. In their paper, the authors [5] used Naive Bayes classifier on projects Mozilla, Eclipse, and Gnome to compute the prediction model.

In our paper, we have applied SLR and SVR approach in Rapidminer and Weka for defect prediction. Then the results obtained from the Weka and Rapidminer are compared.

The rest of the paper contains: Section II describes the dataset and metrics used, Section III defines the performance measures used for comparison and describes the results of the regression. Section IV shows the comparison between the results of Weka and RapidMiner. Section V specifies the conclusion and future work.

2. DATASET AND METRICS

In this paper, the subsystem "eclipse/cdt/build" of eclipse was selected for the study. The data was collected from the Github repository for eclipse. For each subsystem, the normalized entropy and number of defects are listed in Table I.

Table 1
Normalized Entropy and Defects

Year	*Defects*	*Normalized Entropy*
2015	987	0.719
2014	99	0.204
2013	289	0.235
2012	848	0.449

HCM metrics as proposed by Hassan [4], are then calculated for the subsystem. SLR and SVR is then performed by Weka and RapidMiner for predicting faults using HCM metrics. You can refer [4], to familiarize yourself with the HCM metrics. The next section describes the performance measures used for comparing the results of the two tools.

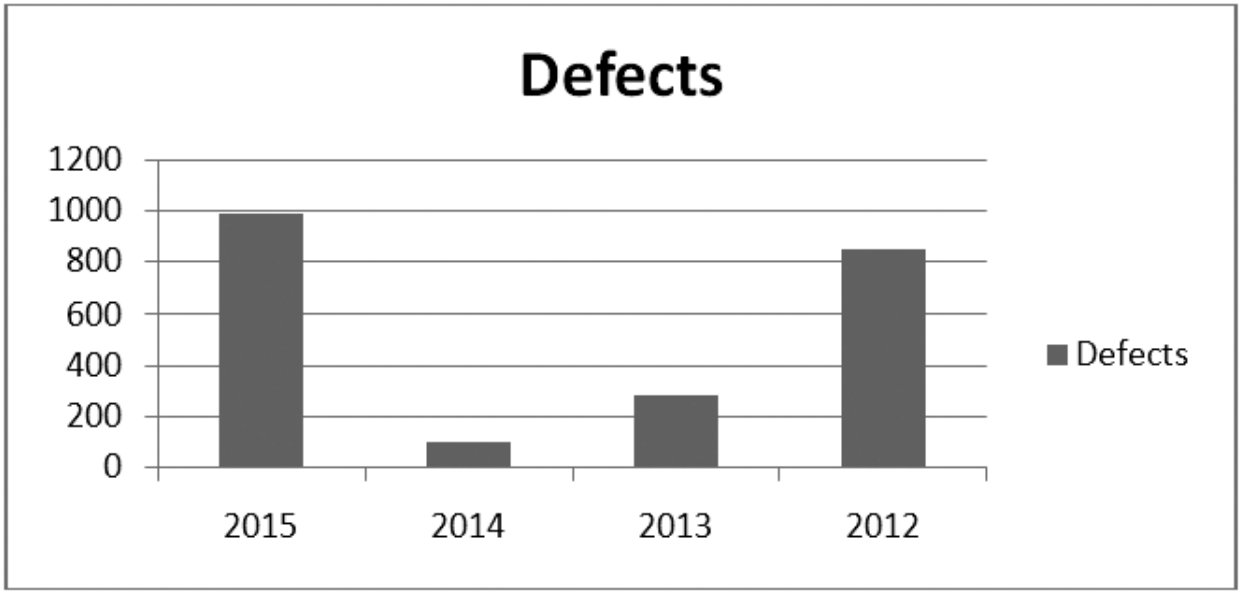

Figure 1: Defects per year

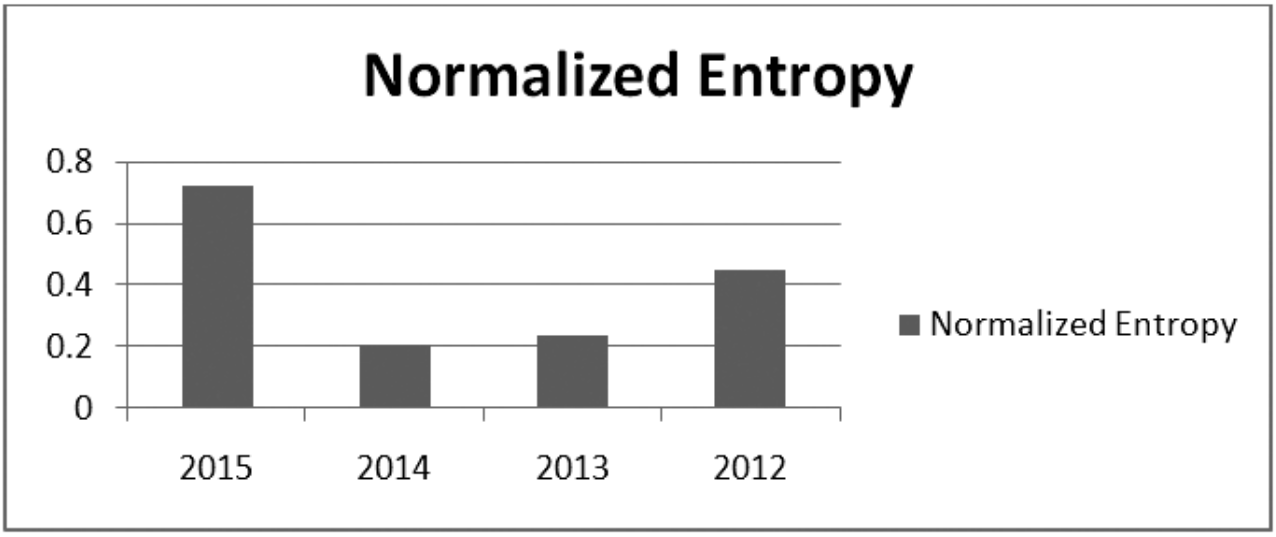

Figure 2: Normalized Entropy per year

3. PERFORMANCE MEASURES AND RESULTS

The performance measures used to evaluate the performance of regression techniques of RapidMiner and Weka are:

A. Correlation Coefficient shows the relationship between two variables. The greater the value better the relationship.

$$\text{Correlation Coefficient} = \frac{Gov(a_x, a_y)}{\sigma_x \ \sigma_y} \tag{1}$$

B. The mean absolute error (M.A.E.) measures how much error we can expect from the prediction on an average. Lesser the value of M.A.E. the better is the technique.

$$M.A.E. = \frac{1}{m}\sum_{i=1}^{m}|a_i - b_i|$$

$$= \frac{1}{m}\sum_{i=1}^{m}|X_i| \tag{2}$$

where absolute error, $|x_i| = |a_i - b_i|$

where $a_1,\ldots,a_m$ is the predicted output, $b_1,\ldots\ldots,b_m$ is the expected output and m is the number of instances.

C. Root mean square error is the square root of mean of the squared difference between the predicted and actual values. Lesser the value of R.M.S.E. better is the performance.

$$R.M.S.E. = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(a_i - b_i)^2} \tag{3}$$

R.M.S.E. and M.A.E. can be compared to determine whether the forecast contains the infrequent errors. The consistency of error size will be less if the difference between the R.M.S.E. and M.A.E. is larger. Performance

measures obtained by using RapidMiner are presented in Table II and those by using Weka are presented in Table III.

Table 2
Rapidminer Results

	SLR			*SVR*		
Techniques used Metrics	*HCM 1*	*HCM 2*	*HCM 3*	*HCM 1*	*HCM 2*	*HCM 3*
Correlation Coefficient	0.964	0.993	0.993	0.964	0.993	0.993
M.A.E	109.843	51.265	51.265	430.476	429.764	429.764
R.M.S.E	137.543	59.351	59.351	517.108	516.171	516.171

Table 3
Weka Results

	SLR			*SVR*		
Techniques used Metrics	*HCM 1*	*HCM 2*	*HCM 3*	*HCM 1*	*HCM 2*	*HCM 3*
Correlation Coefficient	0.96	0.9642	0.9642	0.9606	0.8967	0.8967
M.A.E	7779.90	144.72	144.72	4578.42	233.36	144.72
R.M.S.E	15233.09	177.96	177.96	8878.28	331.64	177.96

4. COMPARISON OF RESULTS BETWEEN RAPIDMINER AND WEKA

The M.A.E., R.M.S.E., and correlation coefficient are calculated for HCM1, HCM2 and HCM3 for each subsystem. Weka is data mining software and contains tools for data preprocessing, regression and classification [7]. Rapidminer gives an integrated environment for data analytics, data mining, and text mining [8]. Regression is used to predict the future values by using the relationship between an independent and dependent variable. SLR and SVR were used with HCM metrics and Number of defects as the dependent and independent variables respectively.

Clearly the correlation coefficient is greater using RapidMiner for both the techniques. Also, M.A.E. and R.M.S.E. values obtained are way less using RapidMiner tool than those obtained using Weka. It is therefore concluded that RapidMiner regression algorithms performed better for defect prediction using HCM metrics [4].

5. CONCLUSION AND FUTURE WORK

In this paper, the two data mining tools Weka and RapidMiner were compared. HCM metrics were calculated for a subsystem of Eclipse. When RapidMiner and Weka are used we noticed positive correlation coefficients using both techniques. In this paper, the tools are compared in terms of their performance measures i.e. M.A.E. and R.M.S.E. On analysis of the results, it was found that RapidMiner gave better results than Weka using both SLR and SVR techniques for defect prediction using HCM metrics. In future, we will focus on improving the performance of the prediction models.

REFERENCES

[1] Kalmegh, Sushilkumar Rameshpant. "Comparative Analysis of WEKA Data Mining Algorithm RandomForest, RandomTree and LADTree for Classification of Indigenous News Data." *International Journal of Emerging Technology and Advanced Engineering* 5.1 (2015): 507-517.

[2] Markov, Zdravko, and Ingrid Russell. "An introduction to the WEKA data mining system."*ACM SIGCSE Bulletin* 38.3 (2006): 367-368.

[3] Singh, V. B., and K. K. Chaturvedi. "Entropy based bug prediction using support vector regression." *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on* IEEE, 2012.

[4] Hassan, Ahmed E., and Richard C. Holt. "The top ten list: Dynamic fault prediction." *Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on*. IEEE, 2005.

[5] Abdelmoez, W., Mohamed Kholief, and Fayrouz M. Elsalmy. "Bug fix-time prediction model using naïve Bayes classifier." *Computer Theory and Applications (ICCTA), 2012 22nd International Conference on* IEEE, 2012.

[6] http://www.cs.waikato.ac.nz/ml/weka/

[7] http://rapidminer.com/

[8] Zimmermann, Thomas, et al. "What makes a good bug report?." *IEEE Transactions on Software Engineering* 36.5 (2010): 618-643.

[9] Anbalagan, Prasanth, and Mladen Vouk. "On predicting the time taken to correct bug reports in open source projects." *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on*. IEEE, 2009.

[10] Abd-El-Hafiz, Salwa K. "Entropies as measures of software information." *proceedings of the IEEE International Conference on Software Maintenance (ICSM'01)*. IEEE Computer Society, 2001.

[11] N. Chapin. An entropy metric for software maintainability. In Proceedings of the 28th Hawaii International Conference on System Sciences, pages 522–523, Jan. 1995.

[12] Zimmermann, Thomas, Rahul Premraj, and Andreas Zeller. "Predicting defects for eclipse." *Proceedings of the third international workshop on predictor models in software engineering*. IEEE Computer Society, 2007.

[13] Myers, Raymond H. "Classical and modern regression with applications (Duxbury Classic)." *Duxbury Press, Pacific Grove* (2000).

[14] Chatterjee, Samprit, and Ali S. Hadi. "Simple linear regression." *Regression Analysis by Example, Fourth Edition* (2006): 21-51.

[15] Gunn, Steve R. "Support vector machines for classification and regression." *ISIS technical report* 14 (1998): 85-86.