Analysis of Prediction Model for Topic based Bug Detection

S Venu Madhav Chitta,*A01964307*

Abstract—**Software metrics has been used to describe the complexity of the program and, to estimate software development time. ”How to predict the quality of software through software metrics, before it is being deployed” is a burning question, triggering the substantial research efforts to uncover an answer to this question. Knowing the locations of future software defects allows project managers to optimize the resources available for the maintenance of a software project by focusing on the most problematic components which can be done by Defect analysis using Predictor models. Defect data analysis is of two types; Classification and prediction that can be used to extract models describing significant defect data classes or to predict future defect trends. In this paper, Principal Component Analysis is applied on the Topic files and thereby analysing Predictive and Explanative powers on different Data sets.**

Keywords—Defect Prediction, Principal Component Analysis, Explanative and Predictive Powers.

I. Introduction

Defect prediction is comparatively a novel research area of software quality engineering. Defects can be defined in a disparate ways but are generally defined as aberration from specifications or ardent expectations which might lead to failures in procedure.Defect data analysis can help us for providing better understanding of the software defect data at large.

A software defect is an error, flaw, bug, mistake, failure, or fault in a computer program or system that may generate an inaccurate or unexpected outcome, or precludes the software from behaving as intended. A project team always aspires to procreate a quality software product with zero or little defects. High risk components within the software project should be caught as soon as possible, in order to enhance software quality. Software defects always incur cost in terms of quality and time. Moreover, identifying and rectifying defects is one of the most time consuming and expensive software processes. It is not practically possible to eliminate each and every defect but reducing the magnitude of defects and their adverse effect on the projects is achievable. Allocating quality assurance resources wisely is a risky task. If some nondefective module is tested for months and months, this is a waste of resources. If a defective module is not tested enough, a defect might escape into the field, causing a failure and potential subsequent damage. Therefore, identifying defect-prone modules is a crucial task for management.

A metric is defined as quantitative measure of the degree to which a system, component or process possesses a given attribute.Applied to software, a metric becomes a software metric. Software metrics play an essential part in understanding and controlling the overall software engineering process.

To construct a prediction model, we must have defect and measurement data collected from actual software development efforts to use as the learning set. In this paper, the dataset is a collection of models and metrics of five software systems and their histories.The five Software systems are namely JDT, Lucene. Pde, Equinox, Mylene. Each software system has different Topic Metrics, Bugs and Base Metrics. The more technical details about the Metrics in Data Sets are discussed in next section. We have presented a Model that applies Principal Component Analysis on various topics thereby calculating the Explanative power and the Predictive power. The lower AIC value better the Explanative power and higher the rcor value better the Predictive power. A software system is viewed as a collection of software artifacts that describe different technical concerns/ aspects. Those concerns are assumed to have different levels of defect-proneness, thus, cause different levels of defectproneness to the relevant software artifacts. We use topic modeling to measure the concerns in source code, and use them as the input for machine learning-based defect prediction models. Preliminary result on Eclipse JDT shows that the topic-based metrics have high correlation to the number of bugs (defect-proneness).

We aim at answering the following research questions:

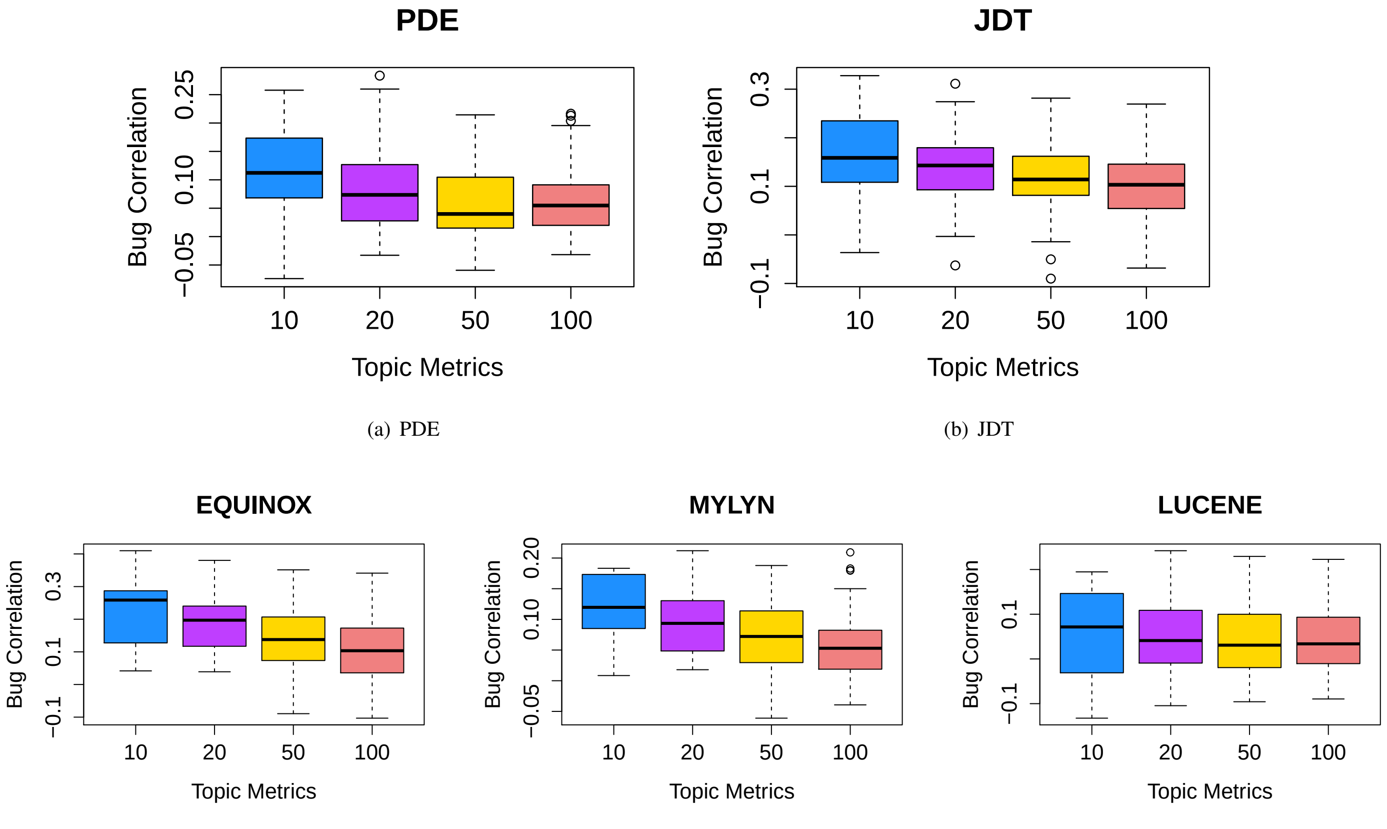
1.What is the correlation of topic metrics to BUG? This question is answered by plotting as boxplots the correlation of BUG and topic metrics for the number of topics of 10, 20, 50, and 100.
2. What are the topics in a dataset provides the best Predictive and Explanative powers for the Prediction Models?

The organization of the paper is as follows: Section 2 explains about the methodology of the design. Section 2A discusses about the the Datasets and Correlation of Topic Metrics with bug. Section 2B the Principal Component analysis is introduced and Discussed. Section 2C discusses about the Explanative and Predictive Powers of the prediction model on various topics of the Dataset. Section 3 discusses about the results that are generated in the design. Section 4 concludes with the final discussion about the results.

II. Methodology

A. Datasets and Correlation of Topic Metrics with bug

The Metrics that are used in this paper are the Topic metrics, Bug and the Base metrics. The Topic metrics has the log transformation of the number of words assigned for each topic (from 1 to K). For example, topic100log.csv will have 100



(a) PDE

(b) JDT

(c) Equinox

(d) Mylyn

(e) Lucene

Fig. 1: Correlation of Topic Metrics with bug

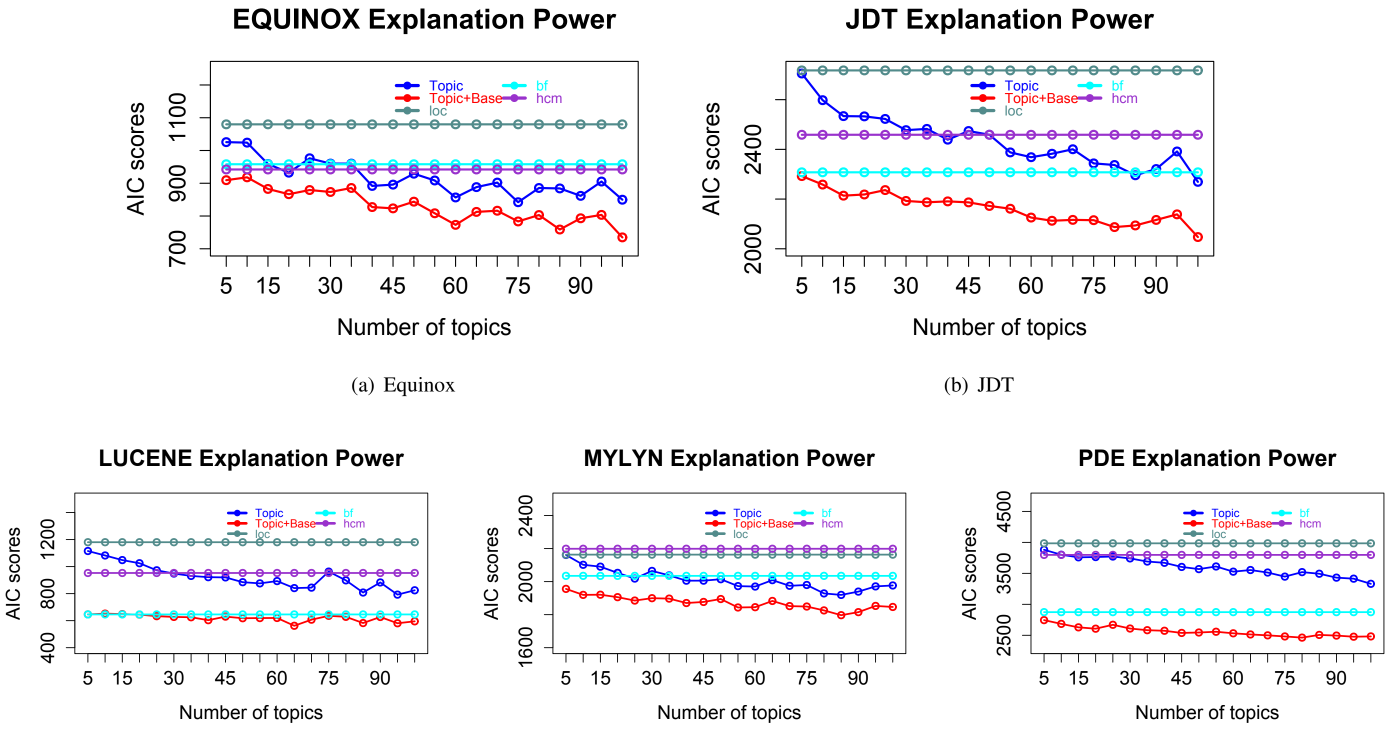
metrics V1-V100, each for a topic. BUG is the actual number of post-release bugs, which should be predicted and not used as predictor. The Base metrics has the metrics which has the properties of the code (LOC, BF, HCM).
LOC: number of lines of code (measuring the code complexity)
BF: number of prior bug fixes (measuring the defect history)
HCM: the entropy of code changes (measuring the complexity of code changes)

The datasets that are analyzed in this paper are : Equinox, Mylyn, JDT, PDE and Lucene. Each dataset contains various topic files. They vary from 1 topic file to 100 topic files. The datasets whose topics are multiples of 5 are analyzed. To answer the research question ”What is the correlation of topic metrics to BUG?”, the correlation between the bugs and the Topic metrics are plotted as Boxplots and then analyzed.

B. Principal Component Analysis

Principal component analysis (PCA) has been called one of the most valuable results from applied linear algebra. PCA is used abundantly in all forms of analysis - from neuroscience to computer graphics - because it is a simple, non-parametric method of extracting relevant information from confusing data sets. With minimal additional effort PCA provides a roadmap for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified dynamics that often underlie it. It uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables.

The major goal of principal components analysis is to reveal hidden structure in a data set. In so doing, we may be able to
1) identify how different variables work together to create the dynamics of the system
2) reduce the dimensionality of the data



(a) Equinox

(b) JDT

(c) Lucene

(d) Mylyn

(e) PDE

Fig. 2: Explanative powers of the metrics in Data Sets

3) decrease redundancy in the data
4) filter some of the noise in the data
5) compress the data
6) prepare the data for further analysis using other techniques

The statistical inferences of Principal Component Analysis are as follows:

PCA principle 1: In general high correlation between variables is a telltale sign of high redundancy in the data.

PCA principle 2: The most important dynamics are the ones with the largest variance.

C. The Explanative and Predictive Powers of the prediction model on various topics of the Dataset

The research question ”What are the topics in a dataset provides the best Predictive and Explanative powers for the Prediction Models” is answered by plotting:

Explanatory power and predictive power of 3 base metrics as baseline.

Explanatory power and predictive power when only topic metrics are used. Plot for K = 5, 10, ... 100 topics are done. For each K, P is varied (number of selected pricipal components) and choose what provides the best explanatory/predictive power.

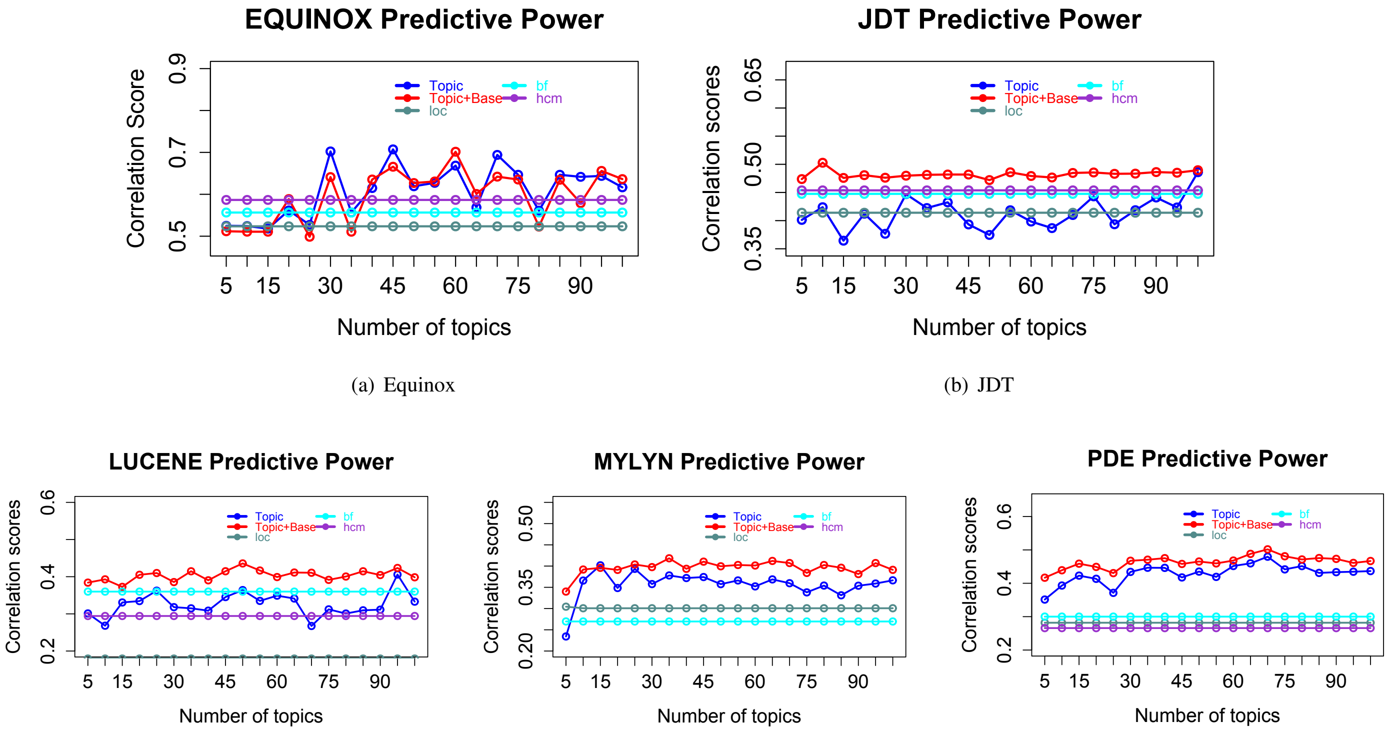
Explanatory power and predictive power when both topic metrics and base metrics are used.Plots for K = 5, 10, ... 100 topics, each with the best P are done

After plotting, the results in tables are summarized, which list the explanatory and predictive power for the best K and P, in addition to those for base metrics.

As explanatory power and predictive power are not comparable, they are plotted in seperate tables and figures and are analysed.

In explanatory power calculation, the AIC values are used. For each topic, you vary number of selected pricipal components and choose the components with low AIC values. To find the best topic, the minimum for the minimum AIC values of each topic metrics ae considered and calculated. Apart from the base metrics, the combined metrics and all the three base metrics are correlated with bug using Linear Regression Model and then best metrics are analysed.

The prediction performance of the PCA components using Linear Regression Model (LM model) We compute Spearmans correlation between the predicted PCA scores and bug metric. Such evaluation approach has been broadly used to assess the predictive power of a number of predictors. In this crossvalidation, for each random split, we use the training set (80of the dataset) to build the regression model, and then we apply the obtained model on the validation set (20producing for each class the predicted number of post-release defects.



(a) Equinox

(b) JDT

(c) Lucene

(d) Mylyn

(e) PDE

Fig. 3: Predictive powers of the metrics in Data Sets

Then, to evaluate the performance of the performed prediction, we compute Spearmans correlation and the mean absolute error between the actual value and the predicted value, on the validation set, between the lists of classes ranked according to the predicted and actual number of post-release defects. Since we perform 30 folds cross-validation, the final values of the Spearmans correlation averages over 30 folds. (4307 is set as the random seed in this paper). Higher the correlation, higher the predictive power.

The prediction and Explanatory powers of the following metrics are considered and the best ones are analyzed.
1) Topic Metrics correlated with bugs.
2) Topic metrics + Base Metrics are correlated with bugs
3) Base Metric (LOC) is correlated with bug.
4) Base Metric (HCM) is correlated with bug.
5) Base Metric (BF) is correlated with bug.

III. Results and Discussion

A. *Correlation of Topic Metrics with bug:*

Fig. (1) shows the correlation of Topic Metrics with bug. The following inferences can be drawn from the figure:

(1) As the number of topics increases, the correlation decreases and they close to 0.
(2) It can be stated that there are too many metrics but not all of them provide necessary information.
(3) By all the above 2 reasons, Method which reduces the metrics with important information is necessary.

TABLE I: Best P and Best K of Explanative Power for each Dataset

Data Set	AIC (Topics)	Best P (Topics)	Best K (Topics)	AIC (Com)	Best P (Com)	Best K (Com)
Equinox	842.13	35	75	734.79	98	100
JDT	2269.30	90	100	2047.23	93	100
Lucene	792.54	84	95	562.09	54	65
Mylyn	1919.41	85	85	1796.79	88	85
PDE	3330.93	95	100	2462.598	81	80

B. *Explanatory Power with Principal Component Analysis:*

The method which is used to resolve the problem as stated in above is Principal Component Analysis. The best Principal Components and best topics for each dataset is provided in table I.

1.Among the five different combination of metrics, the combination of base metrics and Topic metrics provide better explanative power than other four combinations (Base, LOC, BF, HCM).
2. As in Table I, The AIC scores of combined metrics in each dataset is also better than Topic metrics.

C. Predictive Power with Principal Component Analysis:

The best Principal Components and best topics for each dataset is provided in table II.

TABLE II: Best P and Best K of Predictive Power for each Dataset

Data Set	RCOR (Topics)	Best P (Topics)	Best K (Topics)	RCOR (Com)	Best P (Com)	Best K (Com)
Equinox	0.70	17	45	0.70	20	60
JDT	0.48	90	100	0.50	13	10
Lucene	0.40	35	95	0.43	46	50
Mylyn	0.40	4	15	0.41	4	35
PDE	0.47	61	70	0.50	63	70

1.Among the five different combination of metrics, the combination of base metrics and Topic metrics provide better predictive powe rthan other four combinations (Base, LOC, BF, HCM).
2. After PCA the correlation of new metrics are better from original.

IV. Conclusion and Future Work

This paper discusses about Principal Component Analysis which is applied on the Topic files and thereby analysing Predictive and Explanative powers on different Data sets. Results on the datasets shows that the combination of topic and base metrics have higher explanative and Predictive powers than the other metrics. Future work includes applying more statistical methods on the prediction thereby producing the best Predictive model and getting better explanatory and predictive values.

V. References

1. Tung Thanh Nguyen, Tien N. Nguyen, Tu Minh Phuong, ”Topic-based defect prediction: NIER track,” icse, pp.932-935, 2011 33rd International Conference on Software Engineering (ICSE)”, 2011
2 .Nguyen, Anh Tuan, et al. ”A topic-based approach for narrowing the search space of buggy files from a bug report.” Automated Software Engineering (ASE), 2011 26th IEEE/ACM International Conference on. IEEE, 2011.
3. Song, Qinbao, et al. ”Software defect association mining and defect correction effort prediction.” Software Engineering, IEEE Transactions on 32.2 (2006): 69-82.