Predicting Fault in the Process of Producing Important Android Applications using Data Mining Techniques

Asrin Hosseini
M.A student of Computer Engineering
Sanandaj Branch, Islamic Azad University
Sanandaj, Iran

Amir Sheikh-Ahmadi, PhD
Assistant Professor of Computer Engineering unit
Sanandaj Branch, Islamic Azad University
Sanandaj, Iran

ABSTRACT

The aim of this research is to predict fault in the process of producing important Android applications using data mining techniques. Predictable models must not only be correct in predicting fault, but also must be understandable, which needs the user to understand the motivation of the model prediction. Unfortunately, understandability of the fault types is ignored in order to achieve the predictable efficiency of the fault prediction models. In order to solve this problem, some trees are extracted from the random forests and support vector machines for the logistic regression and the rule extraction algorithm are used; also, NASA MDP data are used for extracting the model. The method of evaluating the prediction of software fault is the use of ALPA algorithm and extraction of random trees (RF) and (Logistic) regression for weka software. In the method of creating trees, (REPTree and C4.5) and black box model (Logistic, RF) are used. The results show that the trees extracted from the black box models discuss the prediction way of black box which is more understandable as well as more correct than the trees from direct work on the data.

Keywords

Fault prediction, software effort, Android, data mining

1. INTRODUCTION

During the past decades, there was a revolution in data. Although a lot of information is available in these data, a vast set of their raw data has been hidden. Data mining needs understanding the whole process of extracting these data. Using the complete structure of a fault prediction model in a software company is an accurate and time-consuming one and the commercial projects do not have enough sources for doing so [1]. A software company effort for minimizing the costs is so important. In order to do this, activities such as estimating the software effort and predicting software fault can be very important and can be one of the aims of this research as well. With effort, predicting the software is the aim of the necessary efforts for completing the software project, while the prediction of the software fault tries to determine fault modules. Predicting the software fault enables the software managers to focus the efforts on improving the software quality of the necessary parts. For example, before the system test, identifying those parts which may cause fault while doing the commands can improve the efficacy of the efforts for the software test. So, different methods of modeling the software fault prediction are developed and used for predicting the software quality in reality [2].

Although investigating the prediction and the effort for predicting fault emphasizes the predictable function of a pattern, understandability is a very important aspect which needs the user to understand the reasoning behind the model's prediction. Understandability means how much the users understand the reasoning classifiers and how much the classifiers have a strong mental health. At the time of discussing understandability, there were two main motives for investigation. The first important aspect is the type of output; in the other words, although the ability of understanding the special type of output mostly depends on the domain, the rule-based classifiers and nonlinear classifiers are considered as the most understandable and the least understandable ones, respectively [3]. The second motivation for understandability is the amount of output. In other words, most of the small models are preferred. Understandable models are mostly needed for ensuring the commercial environments and increase the model. Unfortunately, predictable effectiveness and understandability work in a contradictory way and one of them must be ignored because of the other one; in other words, each model has to be adopted in order to achieve the descriptive ad predictable power [4].

In this project, the effort for predicting the fault of the logistic regression method is used for predicting and classifying the software fault. Logistic regression and classifying data mining are predictable in which the target variable is continuous and separated, respectively. Rule extraction depends on a data mining function and its aim is to learn the patterns which happen most of the time [5]. The aim of the research is to find a method for extracting a meaningful rule which can predict a set of software faults. Using the method of rule extraction in this specific field can be useful because the previous researches have shown that in the effort of predicting software fault, nonlinear techniques are usually a better solution for solving these problems [6].

Since many authors have argued the effectiveness of general data for NASA MPD, the achieved results can be useful for the research of software prediction. In addition, it can be seen that the selection of algorithms is done based on the previous researches done in this field. Since selecting the function evaluation is mostly based on the industrial field, the selection of ALPA algorithm can be useful. The methods used are put together based on the common data available in the repository Android and can discuss the way of predicting model by the logistic regression in the extracted trees from black box models with ALPA. So, the simulated algorithm can have a higher accuracy and speed in comparison with the other algorithms and can improve the functional evaluation compared with the simulated algorithms.

This paper includes five parts that are organized as the following: first part: introduction, second part: literature and related works preview, part three: suggested method, part four: evaluation and part five: conclusion.

2. LITERATURE AND RELATED WORKS PREVIEW

2.1 Software Fault

Gear or bag software fault is a type of fault or mistake in running the software which causes incorrect results or stops running the software. The cause of these problems can be the problems of programing. The companies which produce this software provide some versions called beta or alpha before providing the final version in order to be investigated by some people to report these bugs. These people are called Beta Testers.

2.2 Perdition of Software Effort

The aim of evaluating software is to predict the amount of effort needed for completing a software project. A software engineer has to provide human resources, time and budget for each new project, which is unfortunately most of the time challenging. A 2004 CHAOS report has estimated that 53% of the projects are more expensive than the expectation because of the delivery which is not on time and the problems of desired functions which lack the necessity of appropriate evaluation of the amount of needed effort for production and its important features.

2.3 Software Test

Software test means to know if the function of the software is correct or not. With this viewpoint, the test of software products is like the test of other products, but unlike other products, software has different problems. The software users do not see the software problems unless these problems appear on special conditions. On the whole, it can be said that the problems of software are because of their test weaknesses. Due to the increase of complexity and software, the importance of software test increases gradually and the test problems increase gradually, too [7].

2.4 Software Test Methods

Generally, there are two software test methods: black box test and white box test.

2.4.1 Black Box Test

The black box test is a test that does not pay attention to the inner mechanism of a system or tool; it is only focused on the produced outputs based on the selected inputs and running conditions [8].

2.4.1.1 Black box test techniques

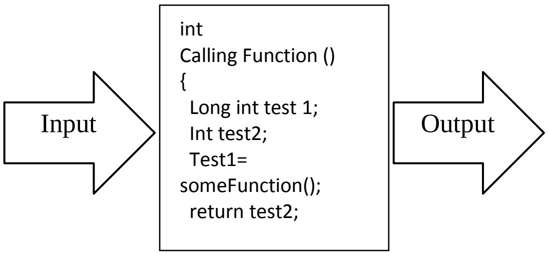
* Analysis of the range amount: This technique is used for decreasing the Test Cases. In this technique, the first and the final amounts are investigated, which means a greater amount than the allowable maximum amount and a smaller amount than the allowable minimum amount will be selected.

* Division: In this technique, data by limiting Test Case are divided into two allowable and un-allowable classes which are both used in the test process.

* Fault guess: In this technique, previous experiences, and human weaknesses are used [9].

2.4.2 White box test

White box test is a test method in which the inner mechanism of a system is tested. Figure (1) shows white box test.



```
int
Calling Function ()
{
  Long int test 1;
  Int test2;
  Test1=
someFunction();
  return test2;
```

Figure1: The test Figure White Box

The white box test figure includes an input, which is analyzed by a stable algorithm, and an output.

The reason why the name of the tests is based on colors is to determine the amount of the test clearness. In the black box test, the software tester does not have access to the application source code. The software is considered as a black box inside of which is not visible. The tester only knows that some data must be entered in the software as the input and the software will produce some output as a result. On the other hand, White box focuses on the inner structure of the code [10].

2.5 Related Works Preview

In a research, Van cruise et al. (2008) focused on the investigating and evaluating the software resources for understandable software and its fault prediction models. The research on the role of ant algorithm (ACO) in the classification method based on AntMier can be the data mining method for predicting the modules of software fault. In the empirical comparison of the general data set in the real world, the rule-based models produced by AntMier are shown for achieving more accuracy in the prediction which is comparable with the other methods of some classification methods such as C4.5, logistic regression and support regression vector. Also, understanding the +AntMier model can be considered in comparing it with the second model [11].

In a research, Binkley et al. (2009) considered the increase of diversity: natural actions for the prediction of the software fault. In this research, some actions are introduced based on three process languages and are used for the problem of the faulty prediction. The first measurement is based on the use of the natural language in the application ID. The second measurement is related to the adaptability of the application ID. The third measurement is considered as the QALP score, which uses the techniques of retrieving information for judging the software quality. The shown QALP score relates to the human judgment of the software quality. Two functions of the language process actions for the software fault prediction were considered by using two applications (open source, specifically). The results shown in the language actions are showed for the process of improving the software fault prediction, especially when they are used in a combination. Generally, the model discusses one third and two third of the software fault in two cases. In the other use of language process, the value of the three actions increases by the amount of the application module [12].

In a research, Arturk et al. (2015) investigated the comparison of some of the methods of soft computation for predicting software fault. In this research, the first program was used for adjusting the neural system of fazi conclusion (ANFIS) for the problem of the software fault prediction. In addition, artificial neural system (ANN) and the method of support vector machine (SVM) are produced for discussing the ANFIS function. Data used in this study are collected from the engineering software resource and the MacCabe criteria are selected. ROC-AUC is used as the function action. The results achieved for the methods of SVM, ANN and ANFIS were 0.7995, 0.8685, ad 0.8573, respectively [13].

3. SUGGESTED METHOD

As the first step, a data set is processed and the data is selected for learning and authenticating a model. According to the fault prediction, the ID and features of a data set are determined by the variance of zero. After data processing, the input selection is done by the method of CFSSUESETEVAL in WEKA. The predictability function of these models is evaluated by their accuracy for the classification and RMSE for the logistic regression. Accuracy is defined as the amount of experimental data points determined by the rules which are divided by the amount of total data points in an experimental set. On the other hand, reading out will compute the ratio of positive cases which are classified correctly. For the logistic regression, RMSE is used in comparison with the real output amount for computing the module's function. Another criterion used for evaluating the models is fidelity. Fidelity is discussed as the achieved amount of the rule set from the white box technique which is similar to the complicated model. It is defined as the data points for the classification in which the rule set and complicated model are adjusted and divided by the total data points in an experimental set. For logistic regression, fidelity is considered as the RMSD of predicting white box and black box. Since rule extraction is used, using white box by the black box in a better way is a necessity. In other words, it is preferred to use the white box in case it has a better function than the black box. This method is a common function in contexts that have used rule extraction [14].

Finally, since the aim of this research is to understand that the rule set produced by C4.5 are provided for using these criteria, the amount of maximum function of regression tree was considered as 5. So the description of the final model is easier ad understandable.

3.1 Techniques of Classification and Logistic Regression

Regression is the support of the vector of a learning step which is on the basis of statistical learning theory. SVMs are produced for solving the classification problems, but they are related to the regression problems, too. So a replacement destruction function is provided which includes distance criterion [15].

3.1.1 Random forest

RF is a similar method which shows a set of classified in different trees [16]. The first step includes randomly sampling L by other N from the main data by the use of BOOT strop sampling. Then, based on the samples, a tree is selected by the random technique of input variables which are selected from the total input variables in the main data (K$\leq m$) [17]. The amount of input variables selects the learner criterion and stay during the tree development process. Although RF is produced on the basis of multiple decision-making trees which are collective, it causes the lack of the profits and produces an output which is difficult for description.

3.1.2 Logistic regression

Logistic regression is a kind of probability, statistical regression model for predicting by classifying suitable information for a logistic curve. Also, for predicting binary response, a binary predictor is used for predicting the results based on one feature or more than one feature.

3.1.3 C4.5

C4.5 is a tree creation technique which is produced based on the information theory concepts and uses entropy for computing the place of a separated data in a special way. Entropy determines the discipline between the observations according to the groups. If $P_1(P_0)$ is considered as the ratio of the examples of group 1(0) of an example, it can be said that entropy equals 1 if $P_1 = P_0 = 0.5$ and equals 0 if $P_1 = 0$ or $P_0 = 0$ when all the observations belong to the same group. In order to decide about the features separated in a related node, the profit criterion is used. Profit is defined as the unexpected decrease in entropy because of the separation of featureX_J. C4.5 uses the criterion of profit ratio and uses normalization for avoiding some of the characteristics of the different features [18]. The decision-making tree is well-known because it is easy and will be easily described if it has a small size [19, 20].

3.1.4 Regression Tree

Regression tree creates a tree in the cubic field by separating input data. Its branches are selected for improving the selected criteria. So that each of its leaves shows one of the fields and each of them uses a simple model [21]. REP tree algorithm, which improves the decrease of the information variance in the data, is used in this research. After learning the tree, the function of the tree decreases with the decrease of fault for generalization. The output of this tree is understandable because the relation between the input and the output is clear.

4. EVALUATION

4.1 Evaluation Criteria

4.1.1 ALPA

In order to increase a set of rules due to the avoidance power or fidelity, one of the rule techniques can be used to determine the output of a more complicated model which acts in a better way. The methods in which such a simulation is understandable are different from the methods of extracting rule, but it is necessary for the function of these techniques. There are three viewpoints in ALPA. First, by providing the amount of predicted training set for the white and black box algorithm instead of the main amount with the training set, the similarity between the white box and black box can be increased. So, the black box becomes a pattern for prediction. Second, since such a pattern is only related to the black box, any new similarity must not be sampled; so, creating new artificial data points and predicting it without any limitations is available. This is one of the details of this algorithm active learning: at the selected point, the algorithm can select each of the vectors of input data from labeling. Third, improving the fidelity data can be done by selecting vectors because more data can be created. The field for reading the data of classification and logistic regression is different.

4.1.2 ALPA classification (ALPAC)

In classification, a model creates some decision-making limitations. These decision-making limitations show passing from one group to the other group [14]. For classification, this field is considered as the best field if the fidelity is improved. The function of the parameter is not clear for RF, but it can be created by creating data in this field through using a proxy based on the unclear function for RF which is described as the following:

$$\pi\big(C_i x\big) = {}^{avg}_{\;\;k} I\big(h_k(x) = y\big)\big) - \underset{j\neq y}{\max}\;\; {}^{avg}_{\;\;k} I\big(h_k(x = j)\big)$$

In which X is the input vector, I() is the indicator function ad H() shows the trees in RF model. By considering this unclear function, those points can be selected that are completely unclear and data can be created by combining these points. So, the artificial data points are placed in a suitable field. Note that the combination of R is placed between two input vectors that are described as:

$$r = \theta x_i + (1 - \theta) x_j, \quad \theta \epsilon [0,1]$$

R is placed on the connecting line of the input axis of the input space.

4.1.3 ALPA regression (ALPAR)

In the regression field, the interesting field is a field that covers the target regression function. In order to create the data around these fields, creating data out of the limitation can be avoided. Unfortunately, the real target function cannot be realized. This can be computed by using the principles of ALPA (ALPAR) classification unless creating the data near the predicted limitation is the goal. With such definitions, it appears near the target function. By using the combination mentioned before, more data points in the correct field can be created which creates the following functional set from ALPA for predicting the software function.

4.2 Results

Table (1) shows the results of the software fault prediction experiments.

Table (1): Results C4.5 compared to ALPAC-trees in terms of Fidelity, Accuracy, Recall and Size.

Data set	C4.5 original				C4.5 ALPAC				RF	
CM1	Fidelity (%)	Accuracy (%)	Recall (%)	Size	Fidelity (%)	Accuracy (%)	Recall (%)	Size	Accuracy (%)	Recall (%)
PC1	85.30	87.00	15.40	5	87.14	81.14	19.20	5	88.95	98.07
KC1	75.99	97.20	47.40	6	82.10	86.10	11	7	93.86	99.21
KC3	81.51	85.60	20.70	7	87.90	87.60	25.46	6	85.44	98.01
Android V2.2	89.22	89.21	22.32	5	89.14	89.68	24.21	7	88.21	25.21
Android V2. 3	87.87	78.22	15.45	8	83.12	83.11	13.85	7	84.34	30.21
Android V4.0	77.16	78.12	19.51	6	78.35	79.54	24.15	5	78.54	45.68
Android V4.2	97.22	94.54	8.16	5	94.38	94.68	17.38	7	93.21	20.05
Wins	3	3	3		5	5	5			

The functions of the target trees and ALPA trees are shown in which the size of each tree is shown in the final column. The results of Random Forest (RF) are provided in the final column of the table. As mentioned before, the accuracy and recall are used for computing the function of the model predicts. Fidelity shows the adjustment of C4.5 ad RF and it is used as an indicator for the amount of rule similarity and complicated model. In other words, the more the fidelity, the better the description of the complicated model by rules. The fidelity, accuracy and recall results show the average test sore 10. In other words, C4.5 has acted in a better way for fidelity and accuracy, a limited number for the better function of the black box is shown in terms of recall. This is considered as a common function in the context of rule extraction, because only the white box model is used if it has a better result and this shows that RF acts better than the target tree for all the data set in terms of accuracy and recall. Compared with ALPAC tree, RF acts better for data sets and it is logistic because the rules are extracted from the black box. In comparing ALPAC tree with the main tree, it can be seen that in 5 to 8 cases, ALPAC acts better in terms of fidelity and C4.5 basis algorithm. ALPAC acts better than accuracy in 5 to 8 cases and for recall acts better for 5 data sets. This can be described as the following: ALPAC is improved for more accuracy than recall. If the aim is to have access to more recall, ALPAC can be used for improving the model for recall. Here, the focus is not on the use of functional criterion but is on the ALPAC potential for creating the understandable rule set of a complicated algorithm. These trees are placed in a way to be created with C4.5. Note that having access to the trees that have the same size is difficult because of the nature of this technique, but the comparison of the average of trees size shows that these models have a similar rule complexity which allows the comparison of the results of the models.

The results for predicting the software effort is shown in table 2.

Table (2): Results REPTree versus ALPAR-trees in terms of Fidelity and Accuracy.

Data set	REPTree original		REPTree ALPAR		Logistic
	Fidelity(RMSD	Accuracy (RMSE)	Fidelity (RMSD)	Accuracy (RMSE)	Accuracy (RMSE)
cocomo81	309.79	9.02	9.78	91.01	96.31
cocomonasa_2	1174.19	11.95	5.33	10.24	10.23
KC2	89.42	82.87	86.54	81.23	81.6
JM1	93.75	81.16	94.22	74.12	74.08
Wins	1	1	3	3	

The results of comparing the target tree and ALPAR tree are shown at the first and second column and the logistic function (logistic regression) is provided at the final column. The size is not mentioned for these trees, because the maximum deep 5 is used for the logistic regression tree ad ALPAR which estimate the aim of creating trees which are understandable. To do this, the regression model fidelity regression can be described as their prediction fidelity; in other words, the less the fidelity, the less the white box is able to describe the black box. With ALPAC, the results show the average 10 experiments. So, as it can be seen in the table 2, logistic acts better than the target trees for all data sets. When comparing the function of ALPAR trees with logistic, it can be seen that they are most of the time less than data sets and their accuracy equals the logistic. Comparing target trees with ALPAR shows that ALPAR acts better than the regression trees in terms of accuracy and function for all data sets. Figure 1 shows one of the logistic regression trees that is created by ALPAR.

This tree is described as follows: if the time limitation is a lot for PU, the amount of the predicted effort will be more. An equal KLOC provides more, effort, too. In contrast, if the KLOC is smaller, the predicted effort relates to NASA; in other words, NASA number 2 will be more improved. Since understanding the relation between the output and the input of this tree is easy, it can be said that this tree is more understandable. So, the user can have more clear viewpoints of the function of the black box.

5. CONCLUSION

In this research, the fault ad effort of software is evaluated and data mining techniques are used. Different kinds of data mining were investigated in this research, such as regression, classification and rule extraction. Regression and classifying data mining functions are predictable, which can be continuous and separated for the target variable, respectively. Rule extraction is a description data mining function and its aim is to learn patterns which happen most of the time. In this research, the main focus was on the logistic regression in order to predict the software effort and classify the software fault prediction. The aim of this research is to determine if the rule extraction creates a set of meaningful principle set and their accuracy in predicting the software effort and fault or not. In an empirical research, it is necessary to consider potential threats for authenticating the results. Its firs probable resource is the pre-process steps that include topics such as losing the amount and selecting the input, which has an important role in the experimental results. While these pre-process steps are used in all data sets, more experiments about the effect of these steps on the results can evoke future researches. Second, data used in the experiments can be considered as the probable resource. In other words, some questions can be asked about the continuity of the data in order to be used by the general filed data. So, the results are compared with similar researches in this field. In addition, most of the authors in the field of using general data sets have discussed NASA MPD and Promise; and it is known that the achieved results are effective in the research and investigation of software computational.

In this research, it is shown that the rule sets are improved in terms of fidelity by using ALPA technique and linear regression, which are most of the time improved because of accuracy and recall. On the other hand, the results showed that rule extraction allows having a clearer viewpoint in the complicated models, because all the extracted trees are easily understandable. This can improve the model acceptance by the final user; so, it is hoped it can facilitate the data mining in the field of software development.

6. REFERENCES

[1] Ostrand, T. and E. Weyuker, on the automation of software fault prediction. 2006.

[2] Khoshgoftaar, T. and N. Seliya, Comparative Assessment of Software Quality Classification Techniques: An Empirical Case Study. 2004. 9(3).

[3] Martens, D., et al., Performance of classification models from a user perspective. Decis. 2011. 51(4).

[4] Verbeke, W., et al., New insights into churn prediction in the telecommunication sector: a profit driven data mining approach. 2012. 218(1).

[5] Baesens, B., et al., Using neural network rule extraction and decision tables for credit-risk evaluation. 2003. 49(3).

[6] Moeyersoms, J., et al., Comprehensible software fault and effort prediction: A data mining approach. 2015. 100.

[7] Weben S, H.W., Kimmich J, Systematic testing of data abstractions based on software specifications. J. Software Testing, Verification and Reliability, 1992. 1(4): p. 39-55.

[8] B. S. Gulavani, T.A.H., Y. Kannan, A. V. Nori, and S. K. Rajamani, Synergy: A new algorithm for property checking. In Proceedings of the 14th Annual Symposium on Foundations of Software Engineering (FSE), 2006.

[9] Korel, B., A Dynamic Approach of Test Data Generation. In IEEE Conference on Software Maintenance, 1990: p. 311-317.

[10] Hayes, J.O.a.J., A Semantic Model of Program Faults. In Proceedings of ISSTA'96 (International Symposium on Software Testing and Analysis), 1996: p. 195-200.

[11] Vandecruys, O., et al., Mining software repositories for comprehensible software fault prediction models. 2008. 81.

[12] Binkley, D., et al., Increasing diversity: Natural language measures for software fault prediction. 2009. 82(11).

[13] Erturk, E. and E. Akcapinar Sezar, A comparison of some soft computing methods for software fault prediction. 2015. 42(4).

[14] Junqué de Fortuny, E., Martens, D, Active Learning-Based Pedagogical Rule Extraction. University of Antwerp, 2014.

[15] Shepperd, M., Kadoda, G, Comparing software prediction techniques using simulation. IEEE Trans. Softw. Eng, 2001. 27(11): p. 1014-1022.

[16] Tan, P., Steinbach, M., Kumar, V, Introduction to Data Mining. Pearson Education, Boston, USA, 2006.

[17] Breiman, L., Random forests. Mach. Learn, 2001. 45(1): p. 2-32.

[18] Quinlan, J., C4. 5: Programs for Machine Learning. Morgan Kaufmann Publish-ers Inc., San Francisco, CA, USA, 1993.

[19] Rokach, L., Maimon, O, Data Mining with Decision Trees. World Scientific Publishing Co, 2008.

[20] Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., Baesens, B, An empiri-cal evaluation of the comprehensibility of decision table, tree and rule based predictive models. Decis. Support Syst, 2011. 51(1): p. 141–154.

[21] Witten, I., Frank, E., Hall, M, Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2011.