Estimating Software Quality with Advanced Data Mining Techniques

Multimethod tool for buliding software fault prediction model

Matej Mertik
Faculty of electrical engineering and computer science,
University of Maribor
Maribor, Slovenia
matej.mertik@uni-mb.si

Mitja Lenič, Gregor Stiglic, Peter Kokol
Faculty of electrical engineering and computer science,
University of Maribor
Maribor, Slovenia

Abstract— **Current software quality estimation models often involve the use of supervised learning methods for building a software fault prediction models. In such models, dependent variable usually represents a software quality measurement indicating the quality of a module by risk-based class membership, or the number of faults. Independent variables include various software metrics as McCabe, Error Count, Halstead, Line of Code, etc… In this paper we present the use of advanced tool for data mining called Multimethod on the case of building software fault prediction model. Multimethod combines different aspects of supervised learning methods in dynamical environment and therefore can improve accuracy of generated prediction model. We demonstrate the use Multimethod tool on the real data from the Metrics Data Project Data (MDP) Repository. Our preliminary empirical results show promising potentials of this approach in predicting software quality in a software measurement and quality dataset.** *(Abstract)*

Keywords-component; Software quality, Multimethod data mining, Supervised learning, Software fault prediction models

I. INTRODUCTION

Data mining is positioned between different research domain as statistics, machine learning, database management and data visualization. It is defined as the process of identifying valid, novel, potentially useful, and ultimately comprehensible knowledge from data, used to help by crucial decision making. Current software quality estimation models often involve use of data mining and machine learning techniques for building a software fault prediction models. In this article we present some of such methods integrated within Multimethod datamining tool, where we achieved better results by building the fault prediction model as with standard machine learning methods. We introduce the Multimethod datamining tool which was developed in laboratory for system design University of Maribor [1], and we present the case study of building the fault prediction model based on the data from the Metrics Data Program Data Repository [12]. At the end we conclude with the discussion and overview of the preliminary results, which shows promising potentials and present the future use and development on the described approach.

II. RELATED WORK

Datamining term with Knowledge Discovery in Databases (KDD) has been defined to describe a variety of techniques for (1) identify nuggets of information or decision-making knowledge in bodies of data, and (2) extracting these in such a way that they can be put to use in the areas of decision support, prediction, forecasting and estimation [2].

Today well-known data mining methods are usually single methods approaches, mostly decision trees, regression trees, cognitive approaches and in some cases genetic algorithms. There are some well known tools on the market, which uses such methods for discovery of knowledge (CART® (Classification and Regression Trees), C5.0 and See5 [4], WEKA). Nearby hybrid method approaches were presented which combine different single methods (classifiers) to overcome the disadvantages and limitations of single methods [5]. For the building of software fault prediction models different models from datamining were used in different studies as regression tree models (cart- least squares, s-plus, and cart-least absolute deviation) [6], Bayesian Belief Networks [7], different neural networks models [8], which showed as a reliable indicators for software quality prediction.

In our study we have adapted and combined some single methods approaches with the Multimethod tool on the real datasets from MDP Repository, where we got promising results. The tool and experiment is presented in the next sections beginning with general overview of some efficient single methods approaches.

III. DECISION TREES, SUPPORT VECTOR MACHINES AND GENETIC ALGORITHMS

A decision tree [4] represents a map of the reasoning process. It can be used to explain why a question is being asked. It is formalism for expressing the mapping from attribute values to classes. Most useful are the top-down induction of decision trees (TDIDT) algorithms, which are using different purity measures for the node splits. The principal problem within TDIDT and other algorithms is an overfitting. That problem can be found not only in decision tree based classifier but also in other induction algorithms. To avoid

Proceedings of the International Conference
on Software Engineering Advances (ICSEA'06)
0-7695-2703-5/06 $20.00 © 2006 IEEE

the overfitting of decision trees the different pruning strategies were introduced. The greatest advantage of decision tree is that the knowledge representation is easily understandable and can be used without the computer. Some advantages of mentioned single methods are summarized in the table 1.

TABLE I. ADVANTAGES OF SINGLE METHODS

Method type	**Disadvantages**	**Advantages**
Decision trees	overfitting	Understandable knowledge
SVM	large number of tunable parameters	work well with large number of features
Genetic algorithms	computational power	handle good in optimization problems

Support Vector Machine (SVM) is next learning algorithm typically used for classification problems, where we are dealing with large number of features (text categorization, handwritten character recognition, image classification, etc.). SVM maps training data in the "input space" of a high dimensional "feature space". It determines a linear decision boundary in the feature space by constructing the "optimal separating hyper-plane", which distinguishes the classes. The "support vectors" are those points in the input space which best define the boundary between the classes. In the contrast to the decision tree the approach showed to be more appropriate for the problems with large number of features, however representation of SVM output, because of the principle is not understandable.

Genetic algorithms (GA) are adaptive heuristic search methods that may be used to solve all kinds of complex search and optimization problems [9], for which no efficient heuristic method has been developed. They are based on the evolutionary ideas of natural selection and genetic processes of biological organisms. Simulating the principles of natural selection and “survival of the fittest” first laid down by Charles Darwin, genetic algorithms are able to evolve solutions to real-world problems. They are often capable of finding optimal solutions even in the most complex search spaces or at least they offer significant benefits compared to other search and optimization techniques. Therefore we can find them integrated within previous described standard single method techniques [10].

IV. HYBRID APPROACHES AND MULTIMETHOD

The hybrid approaches rest on the assumption that only in the synergetic combination of single models can unleash their full power. Each of the single method has its advantages as also inherent limitations and disadvantages, which must be taken into account when using the particular method. In general the hybrids can be divided according to the control flow into four different categories. More can be found in [11].

The hybrid systems are commonly static in the structure and cannot change the order or sequence of application of single method. While studying mentioned approaches the idea of hybrid approaches within evolutionary algorithms was proposed. Both approaches are very promising in 1) achieving the goal to improve the quality of knowledge extraction (building prediction models) and 2) are not inherently limited to suboptimal solutions.

Multimethod approach adopts some ideas of hybrid approaches and evolutionary algorithms in a Multimethod framework, which therefore enables:

- Different representation of knowledge (prediction models); combination of decision trees with different purity measures for the node splits and different strategies of pruning, combination of SVM, decision trees and genetic algorithms
- Required transformation of knowledge between different predictions models (single methods).

Because of the standardization process of the methods knowledge representation we have achieved interchange ability and in general greater modularity. Therefore other existing methods cannot be directly integrated and have to be adjusted for the Multimethod framework-standardized representation.

To find a way to enable dynamic combination of methodologies to the somehow quasi unified knowledge representation, multiple equally qualitative solutions was used in the Multimethod, similar to GA approach. We introduced a population composed out of individuals/solutions that have a common goal to improve their classification abilities on a given environment/problem. Transitions between method’s knowledge representations are introduced on individual operator level. All of different methods dynamical joined in the process finding an optimal combination with unified knowledge representation, are represented in the result as different final decision tree (Multimethod tree). Every node in the tree consists of the different single methods and operators and is represented in a user-friendly user interface within the browser (Figure 1, 2).

V. BULIDING SOFTWARE FAULT PREDICTION MODEL

The Multimethod datamining tool was used on the real datasets of the NASA IV&V Metrics Data Program project [12]. The primary objective of the Metrics Data Program (MDP) is to collect, validate, organize, store and deliver software metrics data for the research community. We choose and analyze three different datasets which were compound from product metrics values and the associated error data at the function/method level. The associated error with severity attribute at the function/method level of a module was used as the decision attribute for building a fault prediction model. We build four different prediction models starting with standard single method and Multimethod approach. In the table 2 basic statistics from each dataset are presented.

TABLE II. BASIC STATISTIC FOR DATASETS

Statistics	**Dataset Pc4**	**Dataset Kc3**	**Dataset Kc4**
Number of attributes	44	44	44

All values	16280	2728	10912
Missing values	0	11	179
Instances	370	62	248
Split			
Learn set	79	79	79
Test set	21	21	21

The decision attribute *"severity"* consists from 5 different classes starting from 0 to 5, where 0 represents dangerous errors on the first critical level. Following the results are interpreted and discussed.

VI. Results

With Multimethod tool we generated four different software fault predication models. Each model was build with the different technique: decision tree C4.5 learner, un-pruned and pruned, two different SVM methods and Multimethod technique, which combined mentioned single method techniques within dynamical genetic environment. In every experiment we sampled the accuracy on learning set and test as also the size of the generated model/ classifier. As we can see from the table 3, 4 and 5 the accuracies on the test set on each dataset were significant better in all cases by the Multimethod generated fault prediction model than by other single approaches.

TABLE III. Results on dataset PC4

Dataset Pc4	**Sampled measurements**		
	Size	***Accuracy (Learning)***	***Accuracy (Testing)***
Unprunned C 4.5	116	93,26	67,12
Prunned C 4.5	4	84,17	78,00
Multimethod	18	86,53	79,45
SVM (RBF Kerner)	14	52,52	47,94
SVM (Linear Kerner)	234	93,2	47,94

TABLE IV. Results on datasetKC3

Dataset Kc3	**Sampled measurements**		
	Size	***Accuracy (Learning)***	***Accuracy (Testing)***
Unprunned C 4.5	31	85,71	55,00
Prunned C 4.5	4	73,80	70,00
Multimethod	8	78,57	75,00
SVM (RBF Kerner)	41	90,47	55,00
SVM (Linear Kerner)	21	26,20	20,00

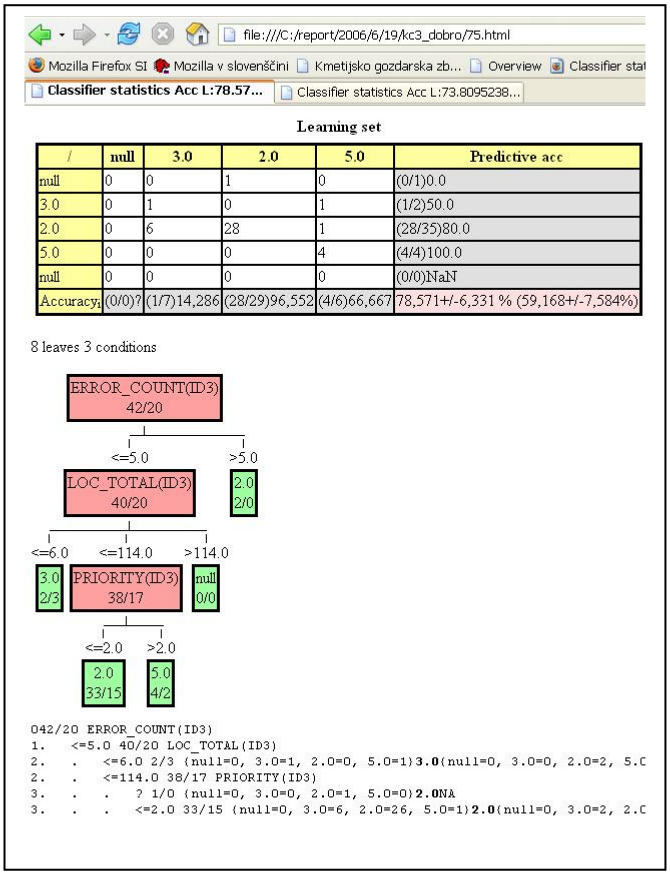
TABLE V. Result on dataset KC4

Dataset Kc4	**Sampled measurements**		
	Size	***Accuracy (Learning)***	***Accuracy (Testing)***
Unprunned C 4.5	89	26,39	19,60
Prunned C 4.5	9	18,27	15,68
Multimethod	3	56,85	45,01
SVM (RBF Kerner)	183	73,60	43,14
SVM (Linear Kerner)	88	38,00	21,56

However we can see that some of single method approaches were better in learning phase; in these cases due to the fact of the parameter size, we faced with the overfitting of a prediction model - classifier. We can see that only in the dataset kc3 the SVM with RBF Kerner was better in accuracies as Multimethod, but SVM produced model is not understandable by experts, what is not the case by the tree of Multimethod. We can conclude that the Multimethod approach over claimed single method approaches in our case study of building fault prediction model. In the next section we will present some examples of the reasoning based knowledge that we got from the fault prediction models generated with the Multimethod.

VII. Generated Software Fault Prediction Models

On the figure 1 and 2 we can see the generated output from the Multimethod datamining tool. In the figure 1, we can see the Multimethod tree for the kc3 dataset.


file:///C:/report/2006/6/19/kc3_dobro/75.html
Mozilla Firefox SI
Mozilla v slovenščini
Kmetijsko gozdarska zb...
Overview
Classifier stat
Classifier statistics Acc L:78.57...
Classifier statistics Acc L:73.8095238...

Learning set

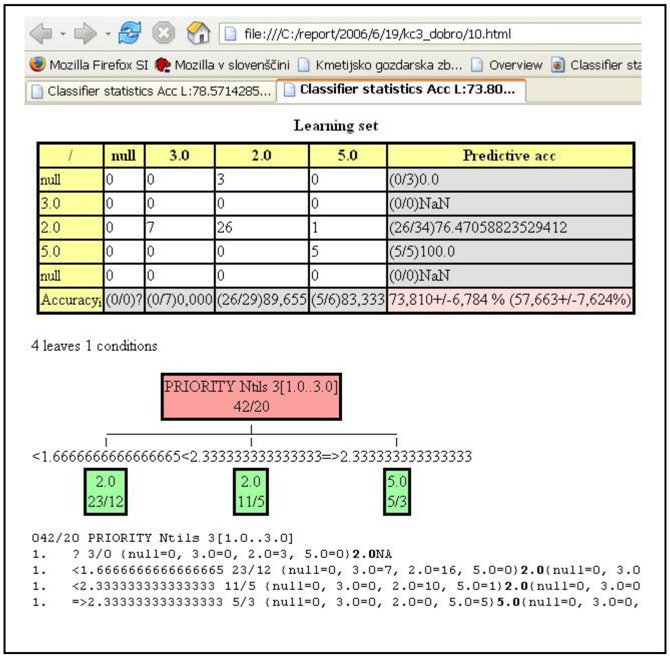
/	null	3.0	2.0	5.0	Predictive acc
null	0	0	1	0	(0/1)0.0
3.0	0	1	0	1	(1/2)50.0
2.0	0	6	28	1	(28/35)80.0
5.0	0	0	0	4	(4/4)100.0
null	0	0	0	0	(0/0)NaN
Accuracy;	(0/0)?	(1/7)14,286	(28/29)96,552	(4/6)66,667	78,571+/-6,331 % (59,168+/-7,584%)

8 leaves 3 conditions

```
042/20 ERROR_COUNT(ID3)
1.   <=5.0 40/20 LOC_TOTAL(ID3)
2.   .   <=6.0 2/3 {null=0, 3.0=1, 2.0=0, 5.0=1}3.0{null=0, 3.0=0, 2.0=2, 5.C
2.   .   <=114.0 38/17 PRIORITY(ID3)
3.   .   .   ? 1/0 {null=0, 3.0=0, 2.0=1, 5.0=0}2.0NA
3.   .   .   <=2.0 33/15 {null=0, 3.0=6, 2.0=26, 5.0=1}2.0{null=0, 3.0=2, 2.C
```

Figure 1. Multimethod extracted knowledge

For the comparison in Figure 2 we are referencing the decision tree generated with pruned C4.5 on the same datasets. As we can see, the extraction of knowledge in represented tree is better by the Multimethod approach as by standard pruned C 4.5.


file:///C:/report/2006/6/19/kc3_dobro/10.html
Mozilla Firefox SI
Mozilla v slovenščini
Kmetijsko gozdarska zb...
Overview
Classifier sta
Classifier statistics Acc L:78.5714285...
Classifier statistics Acc L:73.80...

Learning set

/	null	3.0	2.0	5.0	Predictive acc
null	0	0	3	0	(0/3)0.0
3.0	0	0	0	0	(0/0)NaN
2.0	0	7	26	1	(26/34)76.47058823529412
5.0	0	0	0	5	(5/5)100.0
null	0	0	0	0	(0/0)NaN
Accuracy;	(0/0)?	(0/7)0,000	(26/29)89,655	(5/6)83,333	73,810+/-6,784 % (57,663+/-7,624%)

4 leaves 1 conditions

PRIORITY Ntils 3[1.0..3.0]
42/20
<1.6666666666666665<2.3333333333333333=>2.3333333333333333
2.0
23/12
2.0
11/5
5.0
5/3

```
042/20 PRIORITY Ntils 3[1.0..3.0]
1.    ? 3/0 {null=0, 3.0=0, 2.0=3, 5.0=0}2.0NA
1.    <1.6666666666666665 23/12 {null=0, 3.0=7, 2.0=16, 5.0=0}2.0{null=0, 3.0
1.    <2.3333333333333333 11/5 {null=0, 3.0=0, 2.0=10, 5.0=1}2.0{null=0, 3.0=0
1.    =>2.3333333333333333 5/3 {null=0, 3.0=0, 2.0=0, 5.0=5}5.0{null=0, 3.0=0,
```

Figure 2. C4.5 extracted knowledge

VIII. Conclusion and Discussion

In this paper we presented the advanced datamining tool Multimethod in scenario for building software fault prediction models. We presented use of the tool on three different datasets of the NASA IV&V Metrics Data Program project. The experiments shows promising potentials of using the Multimethod tool for such purposes, as we showed we got quite better results as with standard supervised machine learning methods for building such prediction models like C 4.5 or partly SVM. If we take into consideration the reasoning based knowledge that we get as output in the Multimethod tool, we can conclude that the Multimethod approach is appropriate for building the software fault prediction models because:

- It provides reasoning based knowledge in a form of a Multimethod tree
- It combines different single methods for the building fault prediction software model, what leads to
- Quite good accuracy of predicted model (there are 5 output decision classes)

However our case study was currently implemented on the real data from MDP project. It would be interesting to apply the approach on the other datasets used by the software community, what we are intending to do. We also intent to use the data mining approach of Multimethod with a paradigm of analyzing the software development process in a sense of a complex process, where we are analyzing program complexity using chaos theory [13,14]. We plan to make extensive research on multiple freely available and preparatory projects from industry to confirm or reject our hypothesis, that project complexity and assessment of its state can be made using tools available from chaos theory. For that purposes the presented Multimethod tool for building such models will be used.

References

[1] M.Lenic, Multimetodna gradnja klasifikacijskih sistemov - Phd. Thesis, Maribor 2003.

[2] D.Hand, H.Mannila, P.Smyth: Principles of Data Mining. MIT Press, Cambridge, MA, 2001

[3] L. Breiman J. Friedman, C. J. Stone, R.A. Olshen: Classification and Regression Trees. CRC Press UK, 1983.

[4] J.R.Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann publishers, San Mateo, CA, 1993.

[5] M.Lenic, P.Kokol, “Combining classifiers with Multimethod approach”, V: Second international conference on Hybrid Intelligent Systems, Santiago de Chile, December 1-4, 2002

[6] T. M. Khoshgoftaar, N Seliya: "Tree-Based Software Quality Estimation Models For Fault Prediction," metrics, p. 203, Eighth IEEE International Symposium on Software Metrics (METRICS'02), 2002.

[7] Fenton, Norman, Krause, Paul., Neil, Martin: “A Probabilistic Model for Software Defect Prediction”, IEEE Transactions in Software Engineering, 2001.

[8] M. M. Thwin, T. Quah: “Application of neural networks for software quality prediction using object-oriented metrics”, J. Syst. Softw. 76, 2, 147-156, 2005

[9] D.E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA, 1989

[10] V. Podgorelec, P. Kokol. “Towards more optimal medical diagnosing with evolutionary algorithms”, J. med. syst., 2001, vol. 25, no. 3, page. 195-219

[11] C. J. Iglesias, “The Role of Hybrid Systems in Hybrids”, Control Engineering Practice, Vol. 4, no. 6, 1996, 839-845

[12] World Wide Web: http://mdp.ivv.nasa.gov/index.html

[13] Kokol, P., Podgorelec, V., Zorman, M., Pighin, M., "Alpha - a generic software complexity metric", Project control for software quality (Eds: Rob J. Kusters et al.), Maastricht : Shaker Publishing BV, 1999, pp 397-405.

[14] Cardoso, A.I., Crespo, G., "Is the software process a cahotic one ?", Working paper of Mathematical Science Center, Madeira University,