UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

UNIVERSITY OF SZEGED

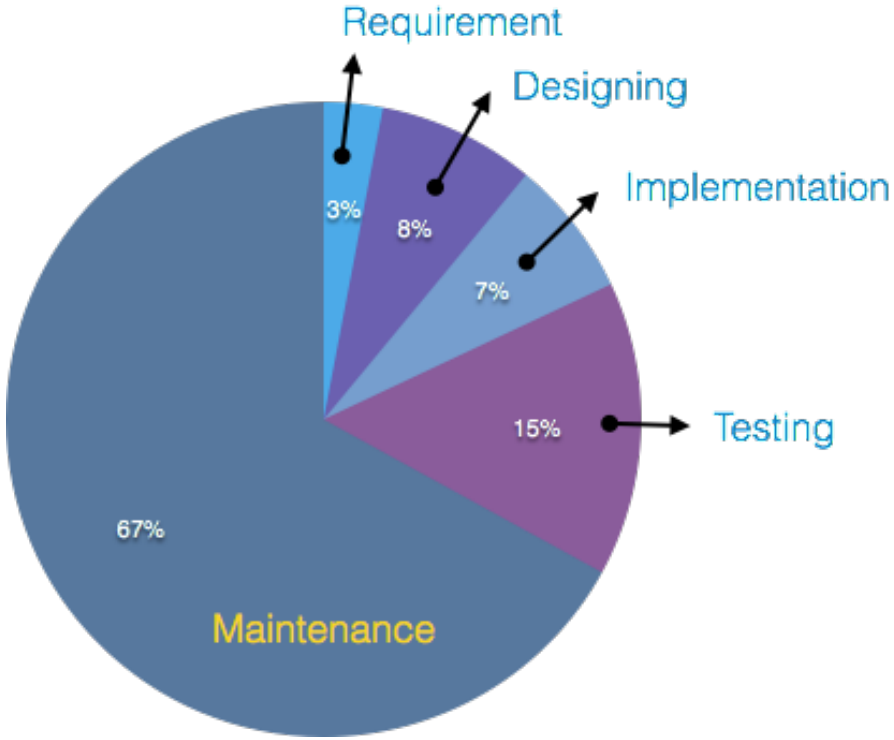
Department of Software Engineering

Assessing Ensemble Learning Techniques in Bug Prediction

Zsolt Szamosvölgyi, Endre Váradi, Zoltán Tóth, Judit Jász, Rudolf Ferenc

Bug Prediction

- With the growing complexities of the software, the number of potential bugs is also increasing rapidly.
- Software bug prediction at the initial stages of software development improves the important aspects such as software quality, reliability, and efficiency and minimizes the development cost.



source: https://blogs.sap.com/2018/01/26/build-better-build-smarter/

Table 1. Metrics calculated by the OpenStaticAnalyzer

Abbr.	Name	Abbr.	Name
AD	API Documentation	NOA	Number of Ancestors
CBO	Coupling Between Object classes	NOC	Number of Children
CBOI	Coupling Between Object classes Inverse	NOD	Number of Descendants
CC	Clone Coverage	NOI	Number of Outgoing Invocations
CCL	Clone Classes	NOP	Number of Parents
CCO	Clone Complexity	NOS	Number of Statements
CD	Comment Density	NPA	Number of Public Attributes
CI	Clone Instances	NPM	Number of Public Methods
CLC	Clone Line Coverage	NS	Number of Setters
CLLC	Clone Logical Line Coverage	PDA	Public Documented API
CLOC	Comment Lines of Code	PUA	Public Undocumented API
DIT	Depth of Inheritance Tree	RFC	Response set For Class
DLOC	Documentation Lines of Code	TCD	Total Comment Density
LCOM5	Lack of Cohesion in Methods 5	TCLOC	Total Comment Lines of Code
LDC	Lines of Duplicated Code	TLLOC	Total Logical Lines of Code
LLDC	Logical Lines of Duplicated Code	TLOC	Total Lines of Code
LLOC	Logical Lines of Code	TNA	Total Number of Attributes
LOC	Lines of Code	TNG	Total Number of Getters
NA	Number of Attributes	TNLA	Total Number of Local Attributes
NG	Number of Getters	TNLG	Total Number of Local Getters
NII	Number of Incoming Invocations	TNLM	Total Number of Local Methods
NL	Nesting Level	TNLPA	Total Number of Local Public Attributes
NLA	Number of Local Attributes	TNLPM	Total Number of Local Public Methods
NLE	Nesting Level Else-If	TNLS	Total Number of Local Setters
NLG	Number of Local Getters	TNM	Total Number of Methods
NLM	Number of Local Methods	TNOS	Total Number of Statements
NLPA	Number of Local Public Attributes	TNPA	Total Number of Public Attributes
NLPM	Number of Local Public Methods	TNPM	Total Number of Public Methods
NLS	Number of Local Setters	TNS	Total Number of Setters
NM	Number of Methods	WMC	Weighted Methods per Class

Traditional Machine Learning methods

- Without any prior knowledge about the location and the number of bugs, managers may not be able to allocate resources in an efficient way.
- Good solution: bug prediction based on software metrics.

Ensemble learning techniques

- Ensemble learning vs. traditional machine learning algorithms
- Practical evaluation: bagging and AdaBoost as ensemble learners.
- Base learners:
  - Slightly better prediction than random choice
  - Gaussian Naiv Bayes and Decision Tree

Bagging

- Bootstrapping: generation of bootstrap training databases uniformly from the original training set
- Parallel training: voting on all bootstrap databases
- Aggregation: majority voting

AdaBoost

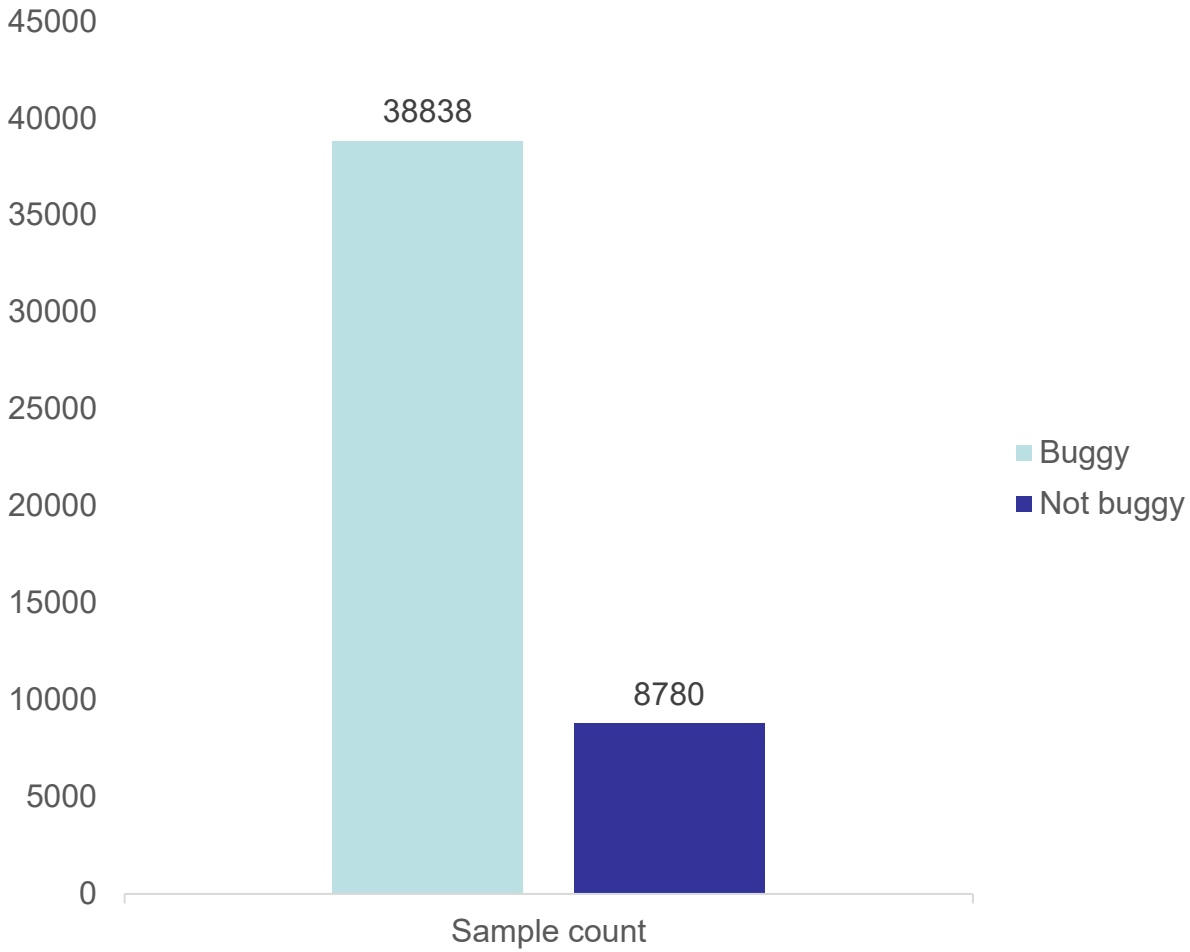
- Adaptive and sequential learning
- most widely adopted ensemble learning method for bug prediction
- final output is the combination of the outputs of the weak learners
- best for decision trees on binary classification problems

Unified Bug Dataset

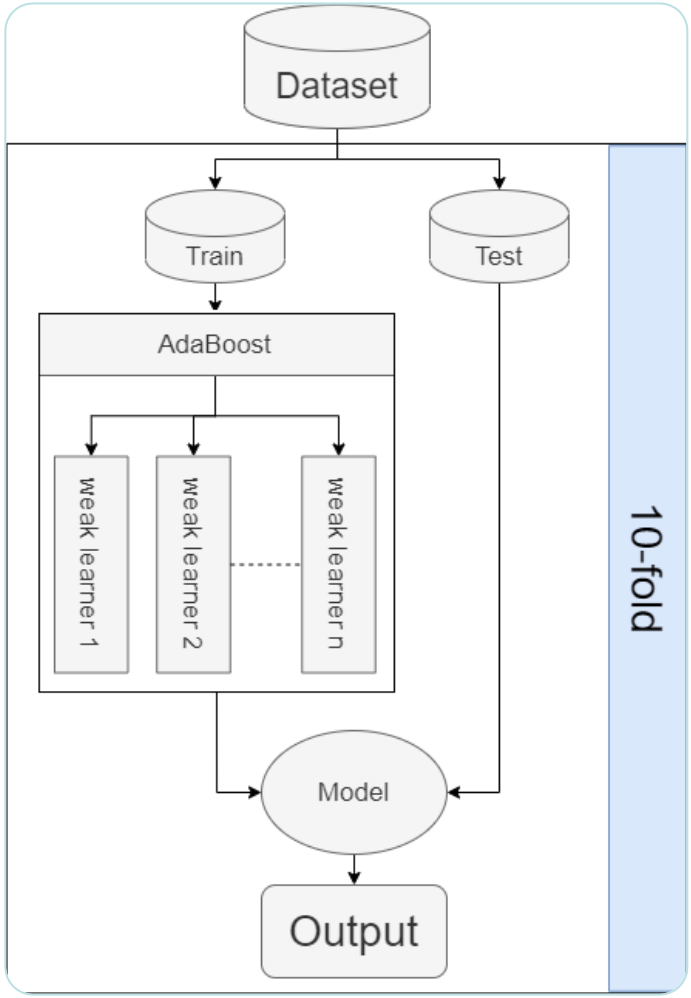
- In order to test the AdaBoost and Bagging classifiers in fault prediction, a large and representative dataset is necessary.
- Our choice is the class-level part of the Unified Bug Dataset.
- The Unified Bug Dataset is an integration of 3 well-known and widely-used datasets.

Resampling methods

- Performing binary classification on an unbalanced dataset is a challenging task.
- Solution:
  - SMOTE
  - RUS
  - RandUp



Evaluation



Accuracy: $$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Precision: $$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall: $$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

F-measure: $$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

RQ1: Does AdaBoost perform better than other classifier methods for bug prediction?

Table 3. Evaluation of ensemble learners

Ensemble learner	Base learner	Number of estimators	Learning rate	F-measure	Accuracy	Precision	Recall
AdaBoost	DecisionTree	100	0.05	54.13%	81.06%	48.91%	60.64%
AdaBoost	DecisionTree	200	0.05	54.61%	81.56%	50.00%	60.17%
AdaBoost	**DecisionTree**	**300**	**0.05**	**54.61%**	**81.96%**	**50.92%**	**58.90%**
AdaBoost	DecisionTree	400	0.05	54.12%	81.89%	50.75%	57.98%
AdaBoost	DecisionTree	500	0.04	53.93%	81.83%	50.62%	57.72%
AdaBoost	**GaussianNB**	**100**	**0.04**	**37.23%**	**77.41%**	**38.38%**	**36.52%**
AdaBoost	GaussianNB	200	0.04	36.10%	75.78%	35.32%	37.23%
AdaBoost	GaussianNB	300	0.04	35.34%	74.36%	33.18%	38.09%
AdaBoost	GaussianNB	400	0.04	34.94%	73.45%	32.04%	38.76%
AdaBoost	GaussianNB	500	0.04	34.58%	72.78%	31.21%	39.09%
Bagging	**DecisionTree**	**100**		**50.54%**	**79.21%**	**45.00%**	**57.67%**
Bagging	DecisionTree	200		50.48%	79.20%	44.98%	57.53%
Bagging	DecisionTree	300		50.48%	79.21%	45.00%	57.49%
Bagging	DecisionTree	400		50.48%	79.22%	45.01%	57.49%
Bagging	DecisionTree	500		50.51%	79.22%	45.02%	57.55%
Bagging	GaussianNB	100		35.14%	80.50%	45.39%	28.69%
Bagging	**GaussianNB**	**200**		**35.16%**	**80.49%**	**45.38%**	**28.71%**
Bagging	GaussianNB	300		35.09%	80.48%	45.33%	28.64%
Bagging	GaussianNB	400		35.12%	80.49%	45.34%	28.68%
Bagging	GaussianNB	500		35.10%	80.48%	45.32%	28.67%

RQ2: Is there any resample technique which performs consistently better than others?

Table 4. F-measure values of different resampling techniques

Ensemble learner	Base learner	None	RandUp.	RUS	SMOTE
AdaBoost	DecisionTree	47.38%	**53.91%**	52.83%	48.01%
AdaBoost	GaussianNB	30.62%	**32.39%**	32.31%	33.21%
Bagging	DecisionTree	28.51%	**50.51%**	49.16%	47.49%
Bagging	GaussianNB	35.02%	**35.10%**	35.53%	36.14%

Table 5. Accuracy values of different resampling techniques

Ensemble learner	Base learner	None	RandUp.	RUS	SMOTE
AdaBoost	DecisionTree	**84.76%**	81.93%	75.27%	84.59%
AdaBoost	GaussianNB	**73.69%**	68.57%	69.15%	63.89%
Bagging	DecisionTree	**83.29%**	79.22%	71.32%	75.72%
Bagging	GaussianNB	80.47%	**80.48%**	80.34%	80.34%

Table 6. Precision values of different resampling techniques

Ensemble learner	Base learner	None	RandUp.	RUS	SMOTE
AdaBoost	DecisionTree	**65.12%**	50.86%	40.75%	63.55%
AdaBoost	GaussianNB	**29.83%**	26.97%	27.23%	25.26%
Bagging	DecisionTree	**67.57%**	45.02%	36.55%	39.50%
Bagging	GaussianNB	45.25%	**45.32%**	44.90%	44.99%

Table 7. Recall values of different resampling techniques

Ensemble learner	Base learner	None	RandUp.	RUS	SMOTE
AdaBoost	DecisionTree	37.26%	57.36%	**75.14%**	38.61%
AdaBoost	GaussianNB	31.49%	40.79%	40.00%	**48.68%**
Bagging	DecisionTree	18.08%	57.55%	**75.18%**	59.56%
Bagging	GaussianNB	28.58%	28.67%	29.41%	**30.21%**

RQ3: Which is the best weak learning algorithm and which parameter configuration is the most powerful?

Table 8. Best hyperparameters for AdaBoost and Decision Tree

Number of estimators	Learning rate	Max depth	Min sample leaf	Criterion
300	0.05	6	22	gini

Table 9. Best hyperparameters for Bagging and Decision Tree

Number of estimators	Max features	Max samples	Max depth	Min sample leaf	Criterion
100	1	1	6	22	gini

Threats to Validity

- The quality of the dataset we used.
- We choose to fix the parameters of the preprocessing techniques, resample techniques and we had a limited search space for our experiments due to our limited resources.

Conclusion

- We focused our study on revealing the capabilities of AdaBoost classifier in bug prediction.
- We concluded that AdaBoost with a proper resampling technique can be an appropriate method for software fault prediction based on static source code metrics, especially combined with Decision Tree classifier. We could achieve 54.61% F-Measure, 81.96% Accuracy, 50.92% Precision, and 58.90% Recall with a proper parameterization.

Ensemble learner	Base learner	None	RandUp.	RUS	SMOTE
AdaBoost	DecisionTree	47.38%	**53.91%**	52.83%	48.01%

Thanks for your attention!