Automated Parameter Optimization of Classification Techniques for Defect Prediction Models

Timothy Dement
North Carolina State University
Raleigh, North Carolina, USA
tmdement@ncsu.edu

ABSTRACT

Recent studies show that defect prediction classifiers may underperform when using default parameter settings, but it is impractical to explore all possible settings in the parameter spaces. The authors of this study have therefore applied Caret, an automated parameter optimization technique, to a case study of 18 datasets from both proprietary and open source domains, and found that (1) Caret improves the AUC performance of defect prediction models by as much as 40 percentage points; (2) Caret-optimized classifiers are at least as stable as classifiers that are trained using the default settings; and (3) Caret increases the likelihood of producing a top-performing classifier by as much as 83%.

1. CASE STUDY OVERVIEW

The authors first performed a literature analysis to find the 30 classification techniques most commonly used for defect prediction and discovered that 26 of these techniques (87%) require at least one parameter setting.

The authors then collected data sets for the study according to the following selection criteria:

C1. Publicly-available defect data sets from different corpora to combat bias and facilitate replication.

C2. Dataset robustness to facilitate stable performance estimates, as determined by an Events Per Variable (EPV) ratio above 10 — i.e., the ratio of the number of occurrences of the least frequently occurring class of the dependent variable to the number of independent variables used to train the model.

C3. Sane defect data to simulate real-world data, as determined by a rate of defective modules below 50%.

An initial 101 data sets from the Tera-PROMISE repository, NASA, and other sources was culled to 18 after applying C2 and C3 criteria, but maintained a variety of source, size, and domain.

The authors then structured the case study with respect the following research questions:

RQ1. How much does the performance of defect prediction models improve when automated parameter optimization is applied?

RQ2. How stable is the performance of defect prediction models when automated parameter optimization is applied?

After observing the large impact that Caret optimization had on both performance improvement and stability, the authors then used their results to revisit prior rankings of defect prediction classifiers.

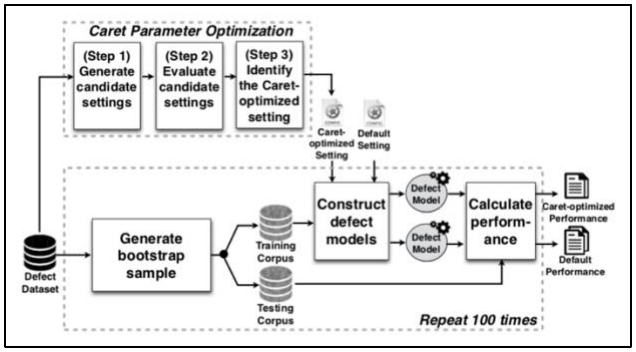
Caret Parameter Optimization
(Step 1) Generate candidate settings
(Step 2) Evaluate candidate settings
(Step 3) Identify the Caret-optimized setting
Caret-optimized Setting
Default Setting
Defect Dataset
Generate bootstrap sample
Training Corpus
Testing Corpus
Construct defect models
Defect Model
Defect Model
Calculate perform-ance
Caret-optimized Performance
Default Performance
Repeat 100 times

Figure 1: An overview of the authors' case study approach.

2. CASE STUDY PROCESS

The authors employed the process diagrammed in Figure 1 to implement their case study. The out-of-sample bootstrap technique was used both during the sample generation step and while evaluating candidate settings during the Caret parameter optimization step, since it has been shown to produce less bias and more stable performance on highly-skewed datasets when compared to *k*-fold cross-validation. In both of the aforementioned steps, the sampling was repeated 100 times and the average performance was reported.

The Caret parameter optimization step employed the `train` function of the `caret` R package to generate a suggested subset of the available parameters for each given classifier, with a budget threshold of 5 for continuous numeric parameters. All combinations of the candidate parameter settings were then evaluated according to the out-of-sample bootstrap technique described above, and the combination with the highest Area Under the Receiver Operator Characteristic Curve (AUC), i.e., the area under the curve that plots the true positive rate against the false positive rate, was selected as the Caret-optimized setting. AUC was chosen as an evaluation metric due to its lack of reliance on an arbitrarily-selected threshold and its resilience to imbalanced data.

The performance of the Caret-optimized models was then evaluated against the performance of the default models using the out-of-sample bootstrap technique and AUC metric.

3. RQ1 (IMPROVEMENT) RESULTS

The results of the case study showed that Caret improves the AUC performance by up to 40%, with a non-negligible improvement in 16 of the 26 classification techniques (62%) as seen in Figure 2, indicating that parameter settings can substantially influence the performance of defect prediction models.

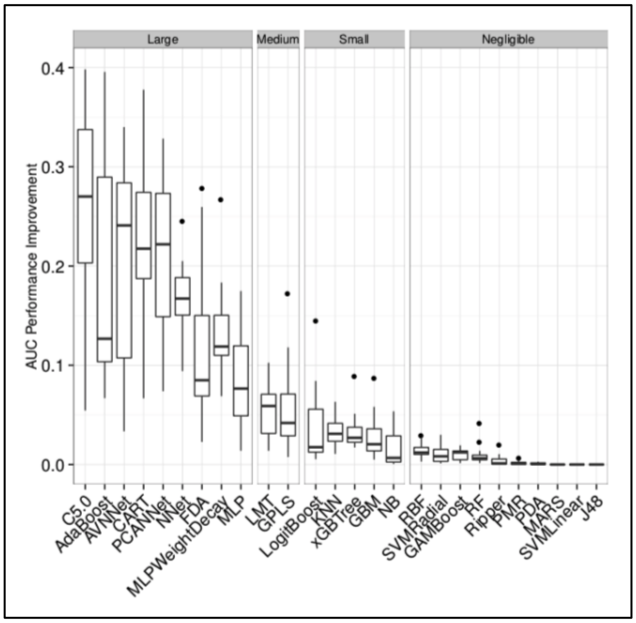

Figure 2: The performance improvement and effect size for each of the studied classification techniques.

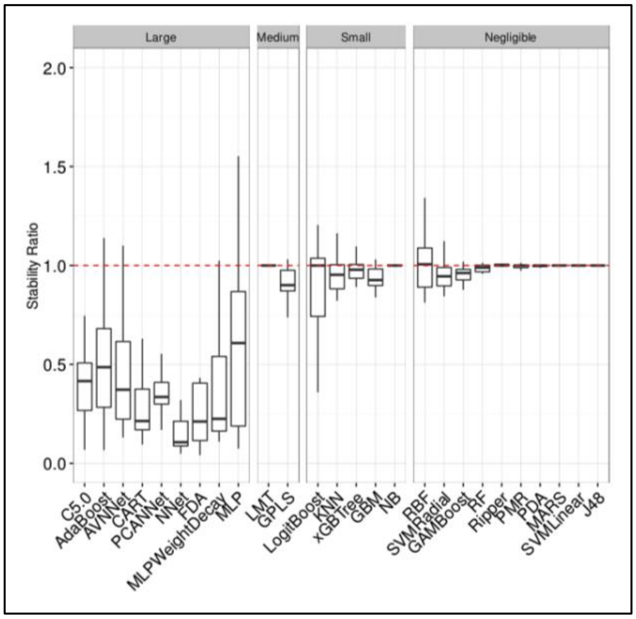

Figure 4: The stability ratio and effect size of the Caret-optimized classifiers compared the default classifiers for each of the studied classification techniques.

4. RQ2 (STABILITY) RESULTS

The performance stability of each classification technique was measured in terms of the variability of the AUC distribution produced by the 100 out-of-sample bootstrap iterations and is presented in Figure 4 as the stability ratio, i.e., the ratio of the standard deviation of the Caret-optimized classifiers to that of the default classifiers.

The results of the case study showed that Caret-optimized classifiers were at least as stable as classifiers that are trained using default settings, with 9 of the 26 studied classification techniques (35%) becoming largely more stable through Caret-optimization and gaining 39-89% stability improvement.

5. ADDITIONAL RESEARCH AND RESULTS

In both RQ1 and RQ2, the authors performed additional study to determine the parameters that had the most drastic effect on the performance improvement and stability of the defect models overall. This was done by isolating a single Caret-optimized parameter for each model while keeping all other parameters set to the default, and then using the out-of-sample bootstrap technique, as well as the AUC and stability ratio metrics, to compare the results against those of the default classifiers (figures not included here).

Another large portion of this study involved revisiting prior rankings of defect prediction classifiers in light of the Caret-optimization results. To explore this question, the authors used a bootstrap-based Ranking Likelihood Estimation (RLE) experiment that leverages both effect size differences and aspects of statistical inference.

The ranking was performed on the AUC performance distribution of the 100 out-of-sample bootstrap iterations of each classification techniques with a Scott-Knott Effect Size Difference (ESD) test. Unlike the standard Scott-Knott test, the ESD variation will merge statistically distinct groups that have a negligible effect size difference. This ranking was performed for both the default and the Caret-optimized distributions, as well as for each of the 26 classification techniques and 18 data sets.

The authors then performed a bootstrap analysis 100 times to approximate the likelihood that a technique will appear in the top Scott-Knott ESD rank when default parameters are used, compared to Caret-optimized parameters, across all of the 18 data sets. The results showed that C5.0 boosting tended to yield top-performing defect prediction models more frequently than the other studied classification techniques, with an 83% likelihood of appearing in the top rank when Caret-optimization applied, compared to 0% with default parameters. These results contradict previous findings that the random forest technique is most likely to produce a top-rank defect classifier. These results suggest that automated parameter optimization can substantially shift the ranking of classification techniques (figures not included here).

The authors also performed an additional cross-context defect prediction analysis, where they trained models in one context but tested them in another context, and still found that Caret-optimization improved model performance by up to 30 percentage points.

6. CONCLUSIONS

The authors note that the cost of applying Caret required less than 30 additional minutes of computation in 17 of the 26 studies techniques (65%). The highest computation cost, more than 3 additional hours, occurred in only 3 of the 26 studied techniques (12%). The authors argue that this cost is relatively minimal, and suggest that defect prediction models do not need to be built often and can be run overnight in the worst cases.

In general, the authors recommend that researchers experiment with the parameters of classification techniques, especially given the simplicity and cost-effectiveness of automated techniques such as Caret and the high performance improvement and stability gain possible through their application.

REFERENCES

[1] Chakkrit Tantithamthavorn, Shane McIntosh, Ahmed E. Hassan, and Kenichi Matsumoto. 2016. Automated Parameter Optimization of Classification Techniques for Defect Prediction Models. In *IEEE/ACM 38th International Conference on Software Engineering (ICSE '16)*, pages 321-332. DOI: http://dx.doi.org/10.1145/2884781.2884857